Problem Set 3

Zixuan Yu

Part I: Get familiar with the data

1. Make a table of mother's parity (alive variable). Ideally, we would compare children of nulliparous women to categories of women of parity > 0. However, in this dataset, there are only 19 children from nulliparous women. So, we will create two categories of women: parity ≤ 3 (i.e. 1 to 4 live births) vs. parity > 3 (5 or more live births).

```
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
```

```
setwd("~/Documents/JHU/term3/biostatmethods3/Biostat653")
load("nepal.anthro.rdata")
dat <- nepal.anthro
d1 = dat[,c("id", "alive", "age","wt","fuvisit")]
d1$parity = factor(ifelse(d1$alive <= 4, 0, 1),
            levels = 0:1,
            labels = c("parity ≤ 3",
                       "parity > 3"))
table(d1$parity)
```

```
##
## parity ≤ 3 parity > 3
##        610        390
```

There are 610 records with parity ≤ 3, and 390 records with parity > 3.

```
table(d1$parity)/5 #5 records for each woman
```

```
##
## parity ≤ 3 parity > 3
##        122         78
```

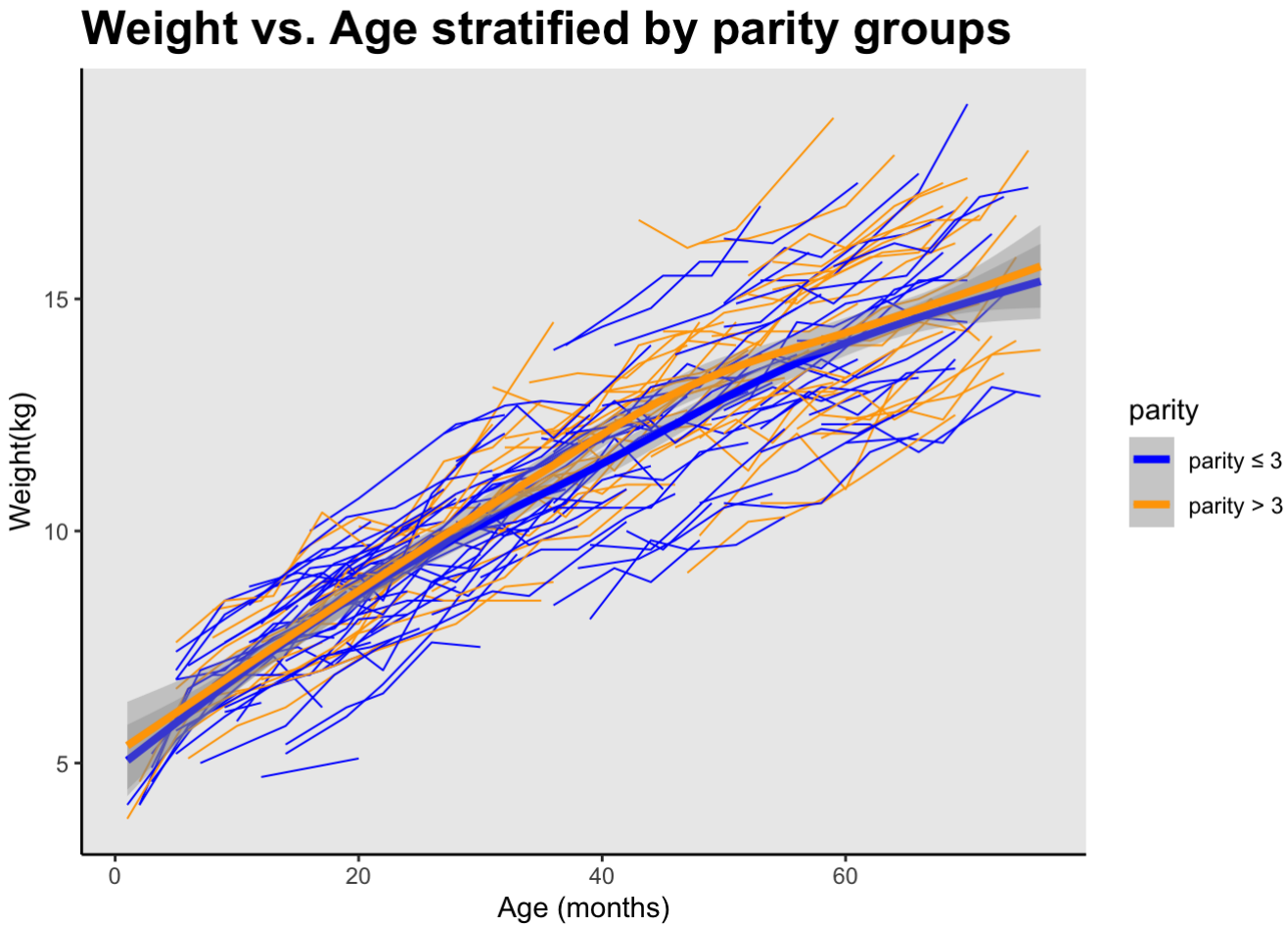
There are 122 children born by mother with parity ≤ 3, and 78 children born by mother with parity > 3. However, there are some missing records, so we have to remove them before plotting/ doing more analysis.

```
d.cc <- d1[complete.cases(d1),]
d.cc = arrange(d.cc,age)
str(d.cc)
```

```
## 'data.frame':    877 obs. of  6 variables:
##  $ id     : int  120681 360471 120361 120381 120411 120291 120384 360571 120691 12
0161 ...
##  $ alive  : int  3 8 5 3 2 4 1 1 3 7 ...
##  $ age    : int  1 1 2 2 2 3 3 3 4 5 ...
##  $ wt     : num  4.1 3.8 4.6 4.1 4.1 ...
##  $ fuvisit: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ parity : Factor w/ 2 levels "parity ≤ 3","parity > 3": 1 2 2 1 1 1 1 1 1 2 ...
```

2.Make a spaghetti plot of children's weight as a function of age; connecting the measured weights within a child over time. Color code the data by parity group. Add smoothing splines for each parity group. Note any similarities or differences in the growth rates across the two parity groups.

```
library(ggplot2)
p1<- ggplot(d.cc, aes(x=age,y=wt, group = factor(id), color = parity)) +
         geom_line(size = 0.35) +
         scale_color_manual(values = c('blue','orange'),
                            breaks = c("parity ≤ 3","parity > 3"),
                            labels = c("parity ≤ 3","parity > 3"))+
         labs(x='Age (months)', y = 'Weight(kg)') +
         geom_smooth(aes(group=parity,color = parity), method = "lm", formula = y ~ s
plines::ns(x,5),se = TRUE, size = 1.4)+
         ggtitle('Weight vs. Age stratified by parity groups')+
         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"))
p1
```



When age is less than 20 months, The parity ≤ 3 group has very similar growth rate as compared to the parity > 3 group. When age is greater than 20 months, the parity > 3 group has larger growth rate as compared to the parity ≤ 3 group. The 'spaghetti' of each child is more spread out as age grows.

Part II: Model checking and recommendations

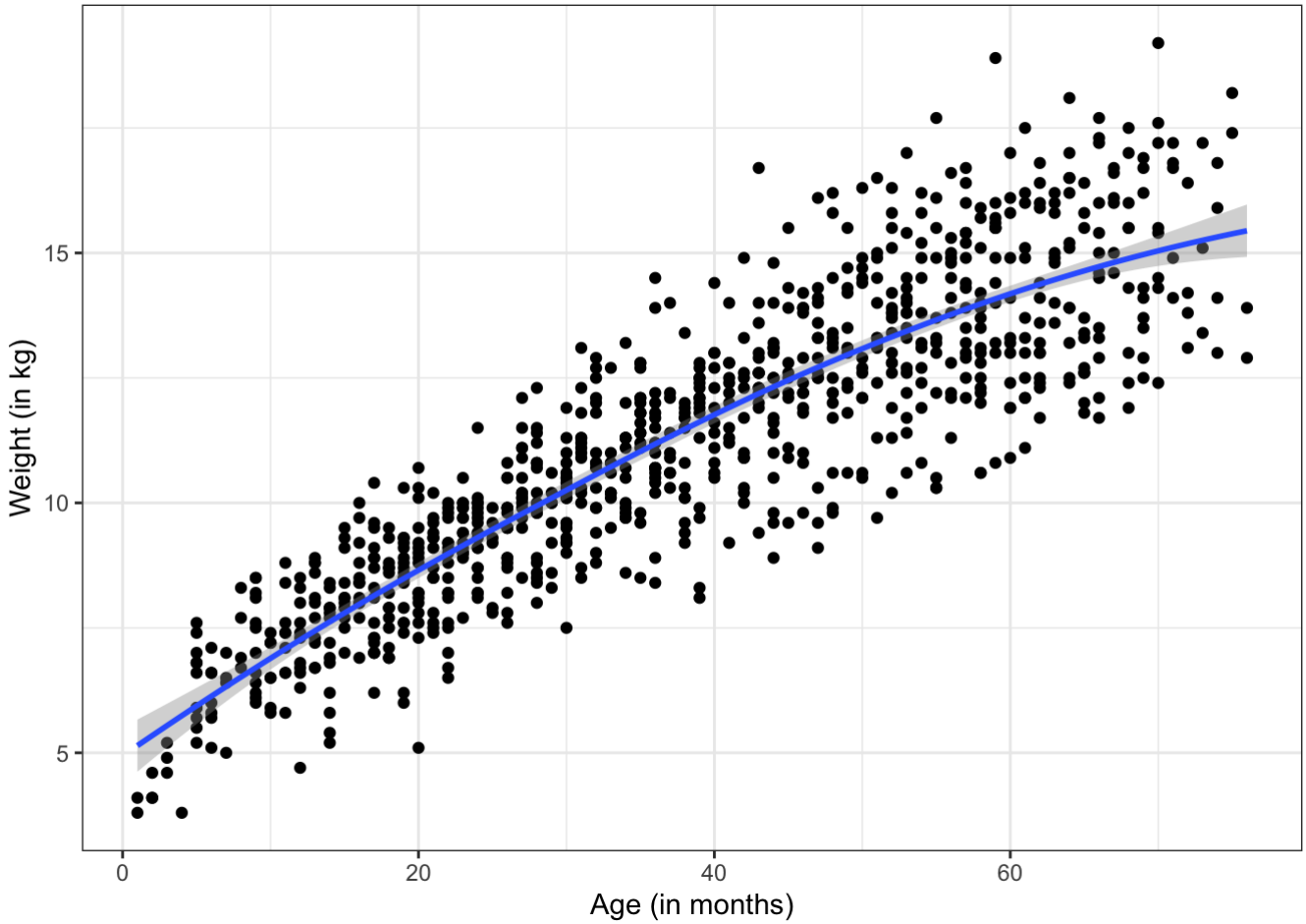
Fit the following model to the data:

$$Y_{ij} = \beta_0 + \beta_2 age_{ij} + \beta_2 I(parity_i > 3) + \beta_3 I(parity_i > 3) age_{ij} + \epsilon_{ij}, \epsilon_{ij}\ N(0, \sigma^2)$$

where i indicates the child $(i = 1, \ldots, 200)$ and j denotes the follow-up $(j = 1, 2, 3, 4, 5)$.
Explore the distribution of weight within each age

```
## I define parity as factor before, in order to fit it into gls model, I create the
 paity_num variable which is numeric
d.cc$parity_num = as.numeric(d.cc$parity)
d.cc$parity_num = d.cc$parity_num-1
model1 <- lm(wt~age+parity_num+parity_num*age, data = d.cc)
ggplot(data = d.cc, aes(x = age, y = wt)) + geom_point() + theme_bw() + geom_smooth()
+ labs(y="Weight (in kg)",x="Age (in months)")  #scale_y_continuous(breaks=seq(2,14,
2),limits=c(1.5,14.5)) + #scale_x_continuous(breaks=seq(0,24,6),limits=c(0,24))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

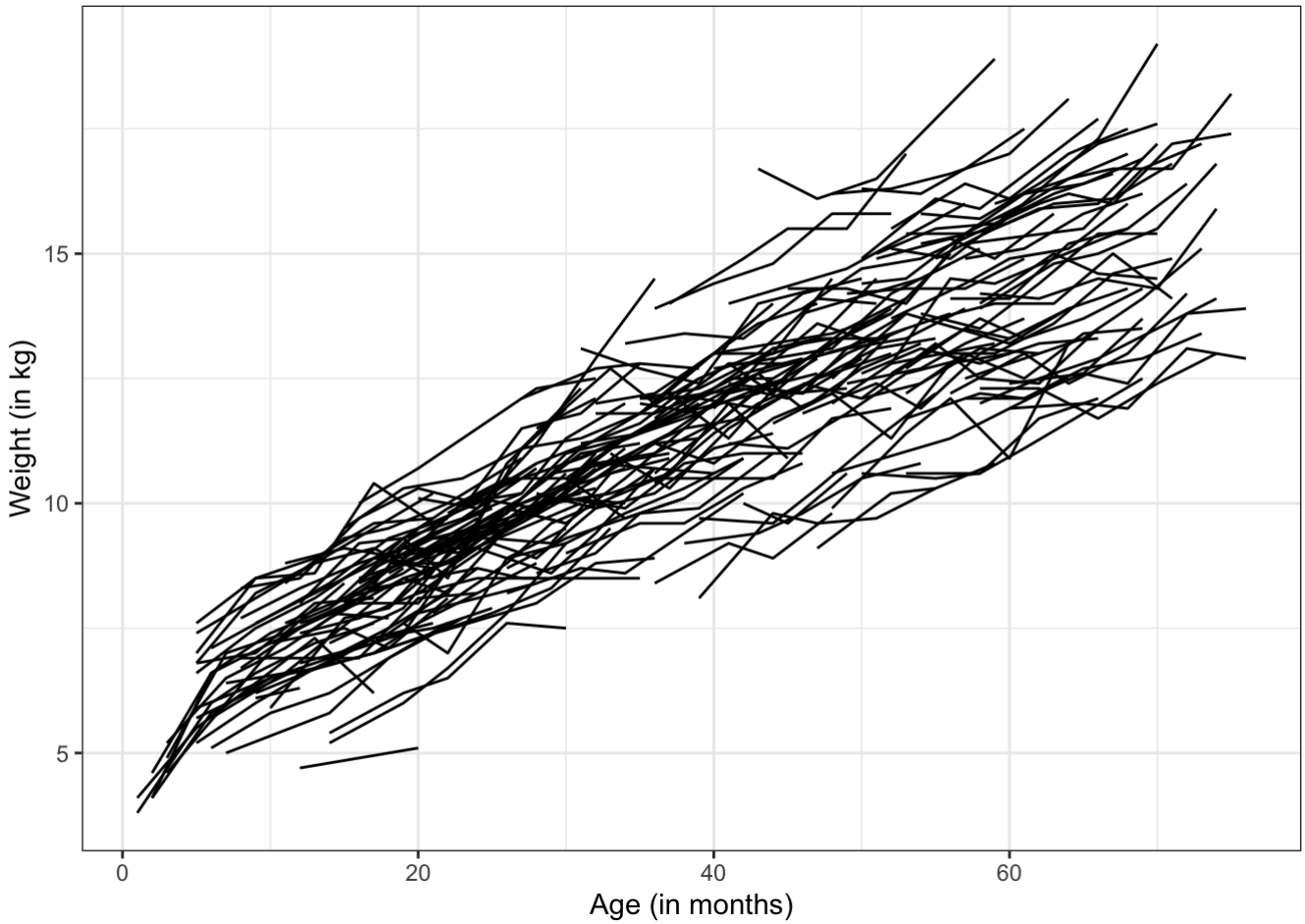



```
summary(model1)
```

```
##
## Call:
## lm(formula = wt ~ age + parity_num + parity_num * age, data = d.cc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5454 -0.9242  0.0818  0.8830  4.6308
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.7843324  0.1367660  42.294   <2e-16 ***
## age             0.1385595  0.0034159  40.563   <2e-16 ***
## parity_num      0.3328985  0.2262539   1.471    0.142
## age:parity_num -0.0001422  0.0052630  -0.027    0.978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.404 on 873 degrees of freedom
## Multiple R-squared:  0.7731, Adjusted R-squared:  0.7723
## F-statistic: 991.3 on 3 and 873 DF,  p-value: < 2.2e-16
```

1.Conduct appropriate checking of this model; i.e. check for appropriateness of the mean model, and the independence and constant variance assumptions for the residuals.

```
## Explore the mean model
ggplot(data = d.cc, aes(x = age, y = wt, group = factor(id))) +
  geom_line() + theme_bw() +
  labs(y="Weight (in kg)",x="Age (in months)")
```

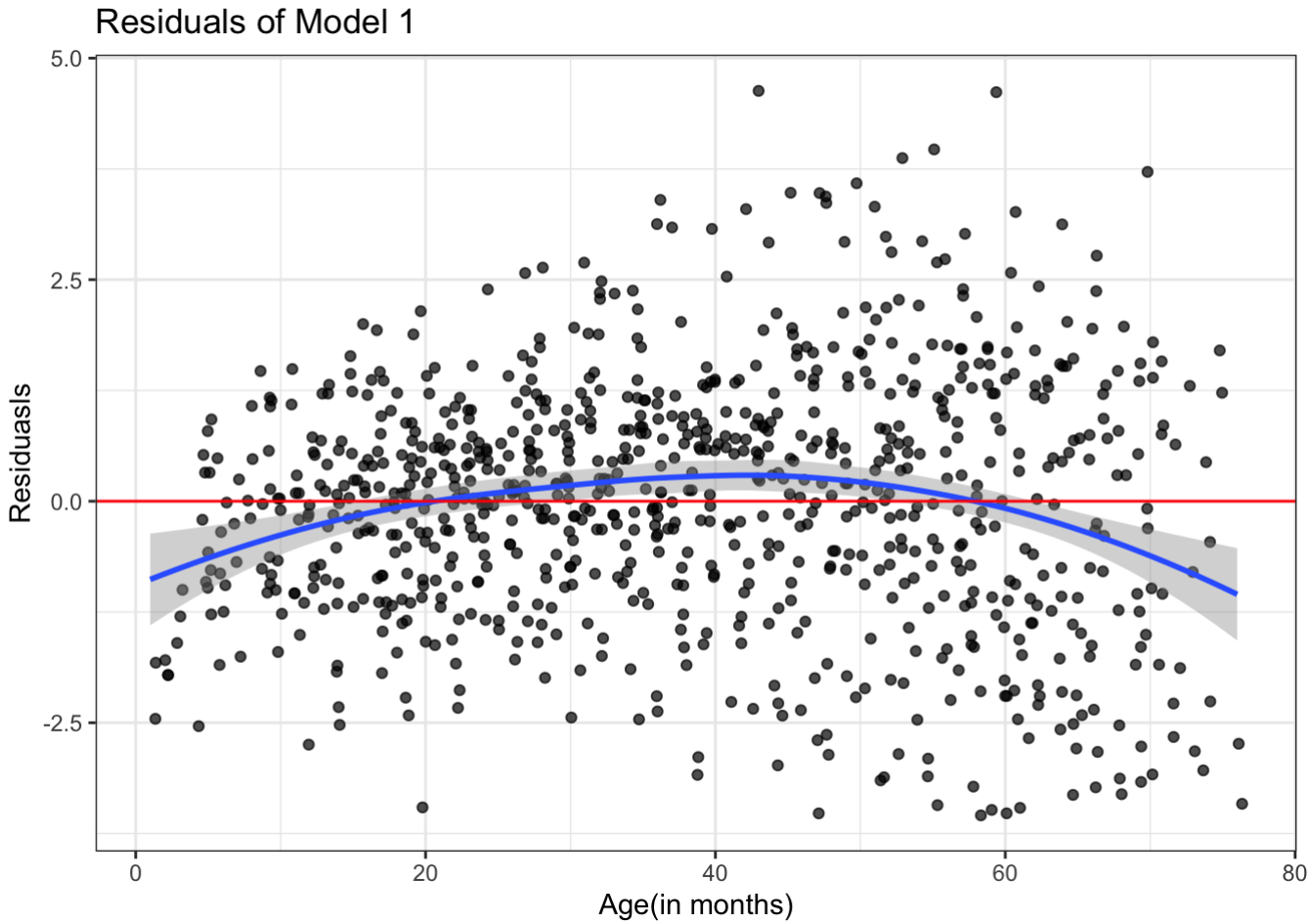



```
  #scale_y_continuous(breaks=seq(2,14,2),limits=c(1.5,14.5)) +
  #scale_x_continuous(breaks=seq(0,60,6),limits=c(0,60))
```

check for constant variance assumption:

```
d.cc$fitval1 = model1$fitted.values
d.cc$res1 = model1$residuals
d.cc$r2 = d.cc$res1^2
p2 <- ggplot(d.cc,aes(x=age,y=res1))+ geom_jitter(alpha=0.7)+ theme_bw()+ geom_smooth
(method = 'loess')+
  geom_hline(yintercept=0,color="red") +
  ggtitle("Residuals of Model 1")+
  labs(y="Residuasls",x="Age(in months)")
#scale_y_continuous(breaks=seq(-3,3.5,0.5),limits=c(-3,3.5)) +
#scale_x_continuous(breaks=seq(0,60,6),limits=c(0,60))
p2
```

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

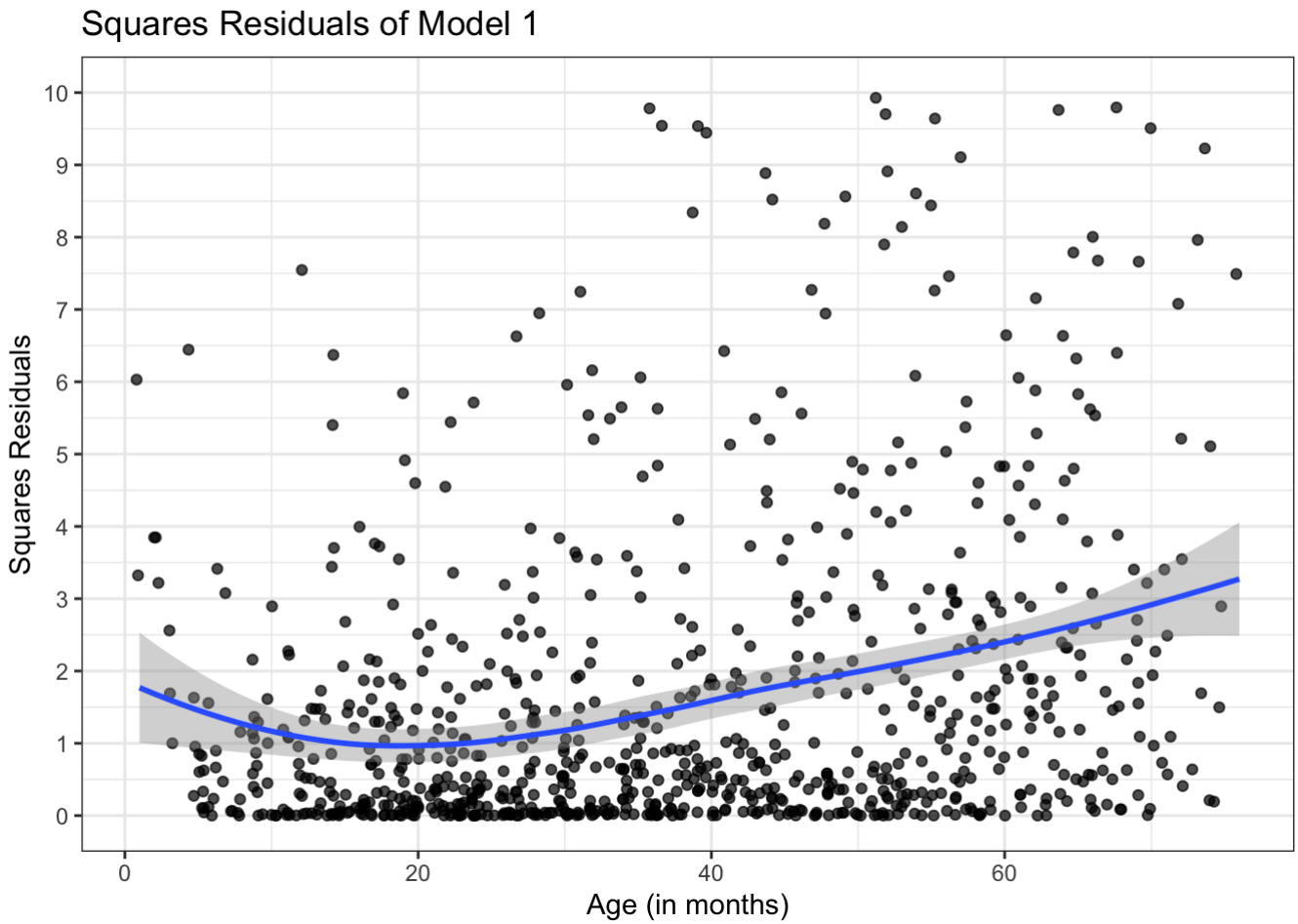



```
d.cc$fitval1 = model1$fitted.values
d.cc$res1 = model1$residuals
d.cc$r2 = d.cc$res1^2
p3 <- ggplot(d.cc,aes(x=age,y=r2))+ geom_jitter(alpha=0.7)+ theme_bw()+ geom_smooth(m
ethod = 'loess')+
  ggtitle("Squares Residuals of Model 1")+
  labs(y="Squares Residuals",x="Age (in months)")+
  scale_y_continuous(breaks=seq(0,10.0,1.0),limits=c(0,10.0))
  #scale_x_continuous(breaks=seq(0,60,6),limits=c(0,60))
p3
```

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

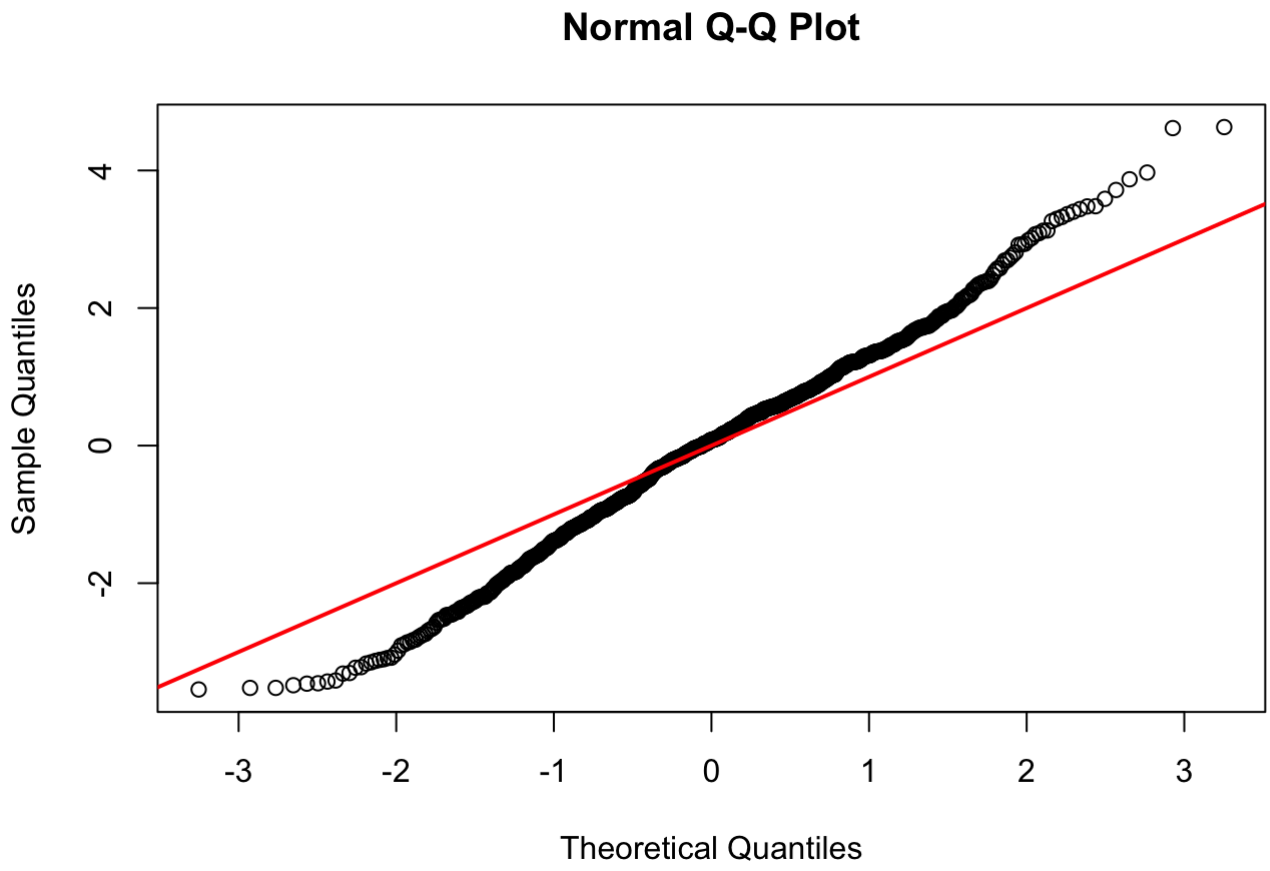
```
## Warning: Removed 27 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 27 rows containing missing values (geom_point).
```



It seems that the constant variant assumption is violated – the squared variance first goes down and reaches its minimum around 20 months of age, then goes up as age becomes larger.

```
qqnorm(d.cc$res1)
abline(0,1,col="red",lwd=2)
```



The normality assumption is greatly violated, the residuals have great departure from the theoretical values. This is because the distribution of children's weight is not normal.

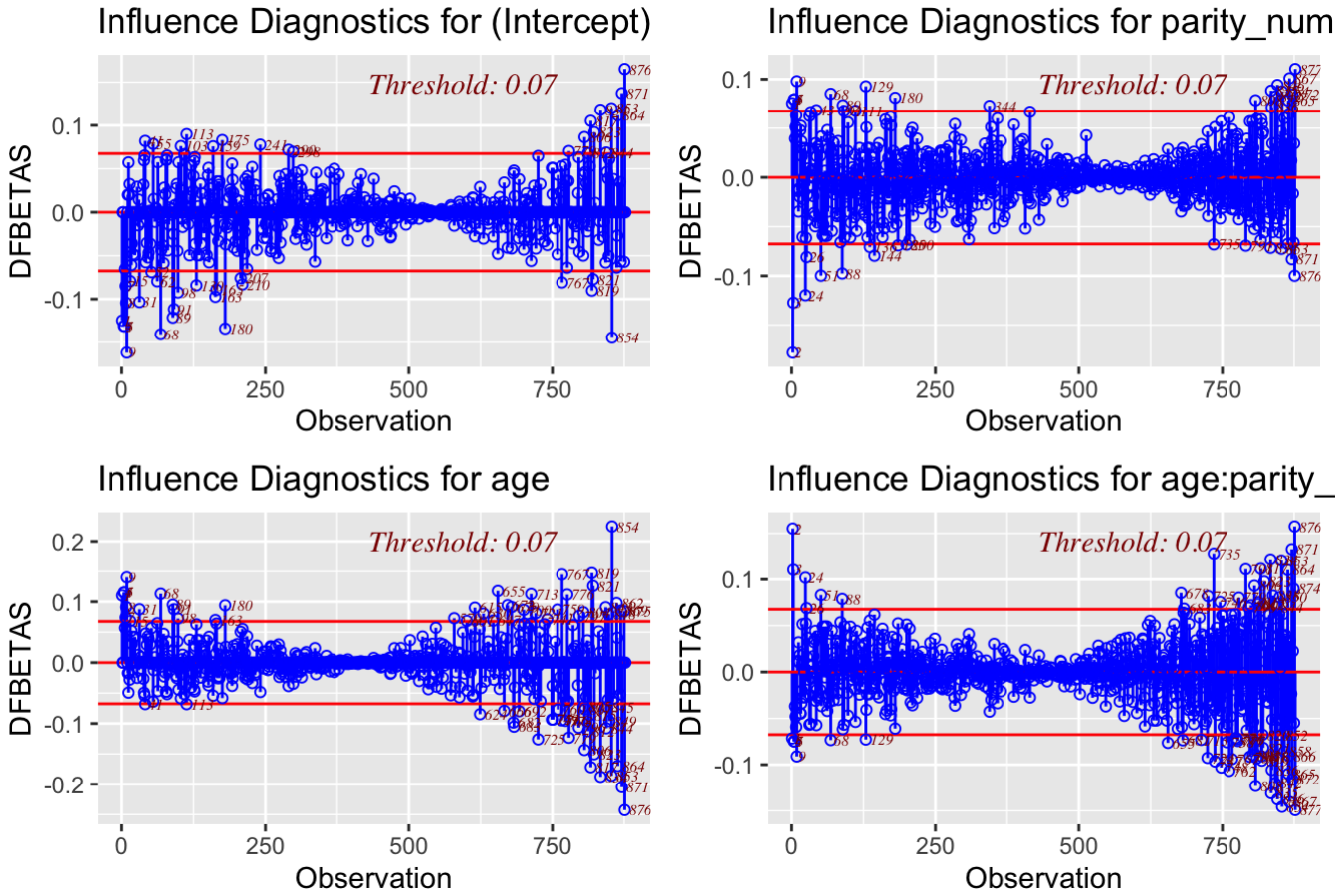
Then Check for Leverage and Influence:
Compute the influence statistics for model 1:

```
library(olsrr)
```

```
##
## Attaching package: 'olsrr'
```

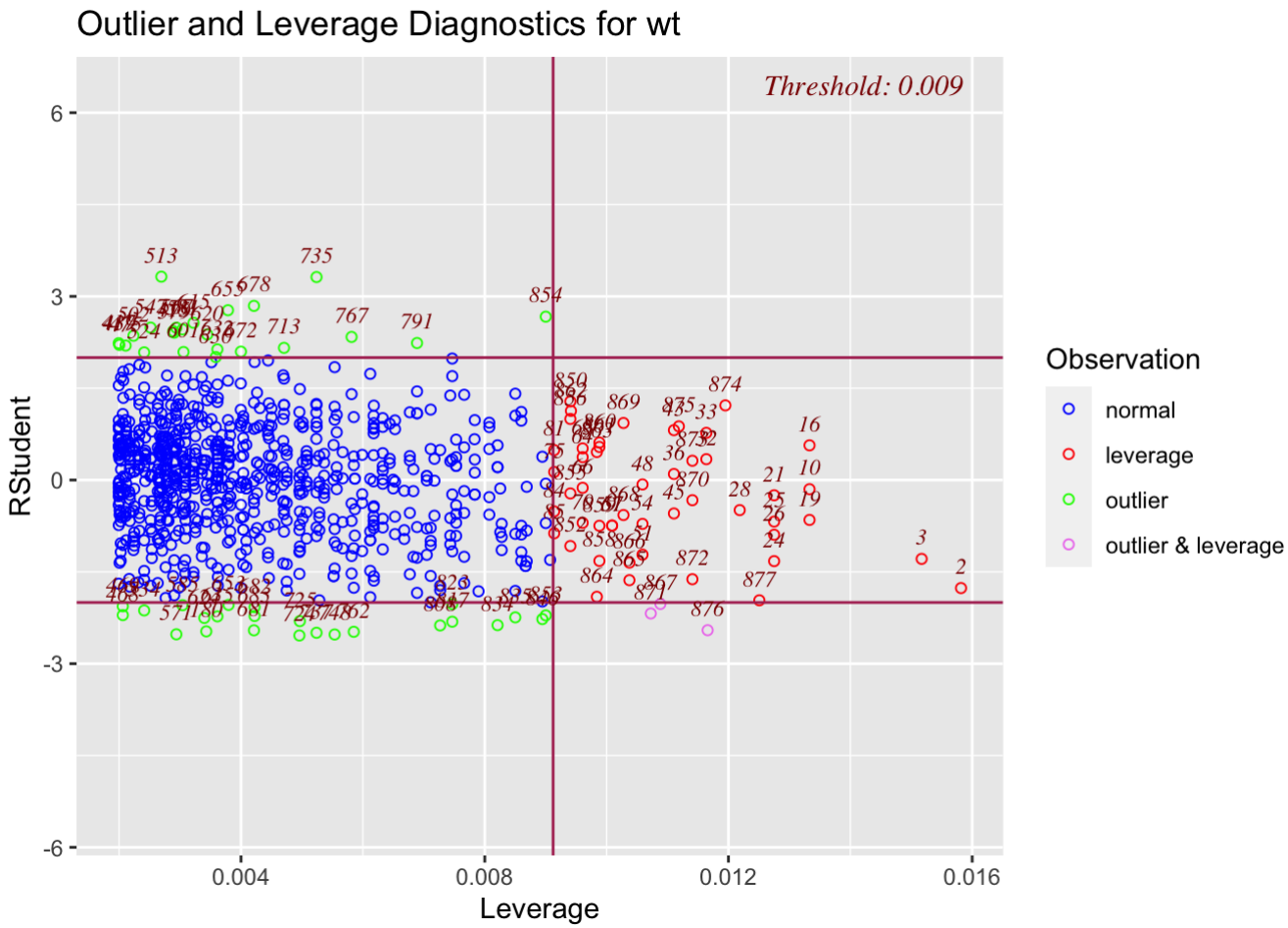
```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
par(mfrow=c(2,2))
ols_plot_dfbetas(model1)
```



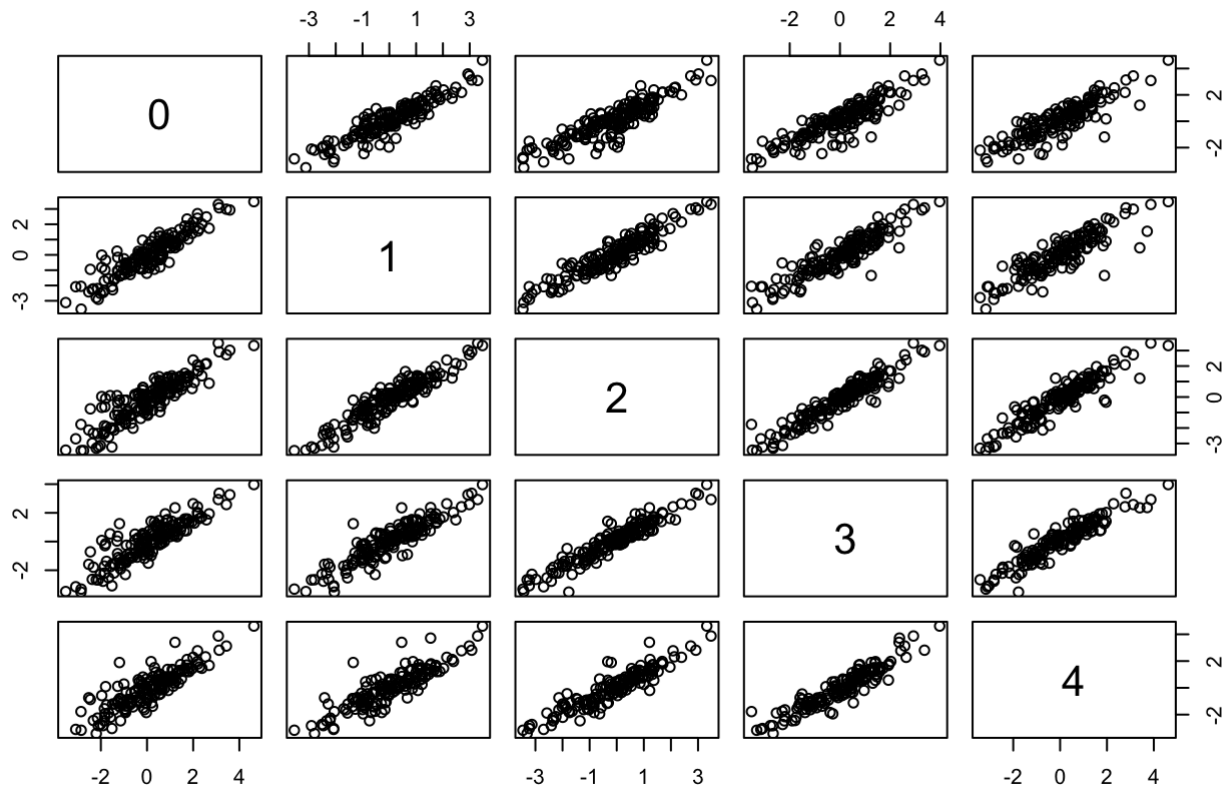
There are many data points with high leverage and influence according to the diagnostic plot.

```
ols_plot_resid_lev(model1)
```



Explore the correlation structure:

```
library(tidyr)
d_wide = d.cc %>% select(id, fuvisit, res1) %>% spread(fuvisit,res1)
pairs(d_wide[,-1], main = "Correlations Structure")
```




```
cor(d_wide[,-1],use = 'pairwise.complete.obs')
```

```
##           0         1         2         3         4
## 0 1.0000000 0.9147354 0.8883770 0.8791622 0.8578079
## 1 0.9147354 1.0000000 0.9378044 0.9070836 0.8723829
## 2 0.8883770 0.9378044 1.0000000 0.9496683 0.9239194
## 3 0.8791622 0.9070836 0.9496683 1.0000000 0.9344300
## 4 0.8578079 0.8723829 0.9239194 0.9344300 1.0000000
```

The independence assumption on residuals are clearly violated. There are strong correlations between residuals.

To sum up, this model is not a good fit for our data. Constant variance assumption and independent error assumptions are violated. Normality of errors is also violated. There are many outliers as well as many data points with high leverage and/or influence.

2.Based on your model checking, propose an alternative model for the data that can address the first goal of the analysis (i.e. determine if the growth rates of children differ by mother's parity (number of previous live births) while satisfying the observed patterns in data with respect to the mean model and distribution of residuals. NOTE: If you modify the mean model, you may want to iterate between model checking for the mean.

Proposed model: Use natural spline terms with degree of freedom of 5 for age.
assume constant variance and non-correlated residuals:

$$Y_{ij} = \beta_0 + \beta_2 ns(age_{ij}, 5) + \beta_2 I(parity_i > 3) + \beta_3 I(parity_i > 3) ns(age_{ij}, 5) + \epsilon_{ij}$$

$$\epsilon_{ij}\ N(0, \sigma^2) Cov(\epsilon_{ij}, \epsilon_{ik}) = 0$$

```
## Proposed Model: Model2
library(splines)
model2 = lm(wt ~ ns(age, df = 5)+ parity_num+parity_num:ns(age, df = 5), data= d.cc)
summary(model2)
```

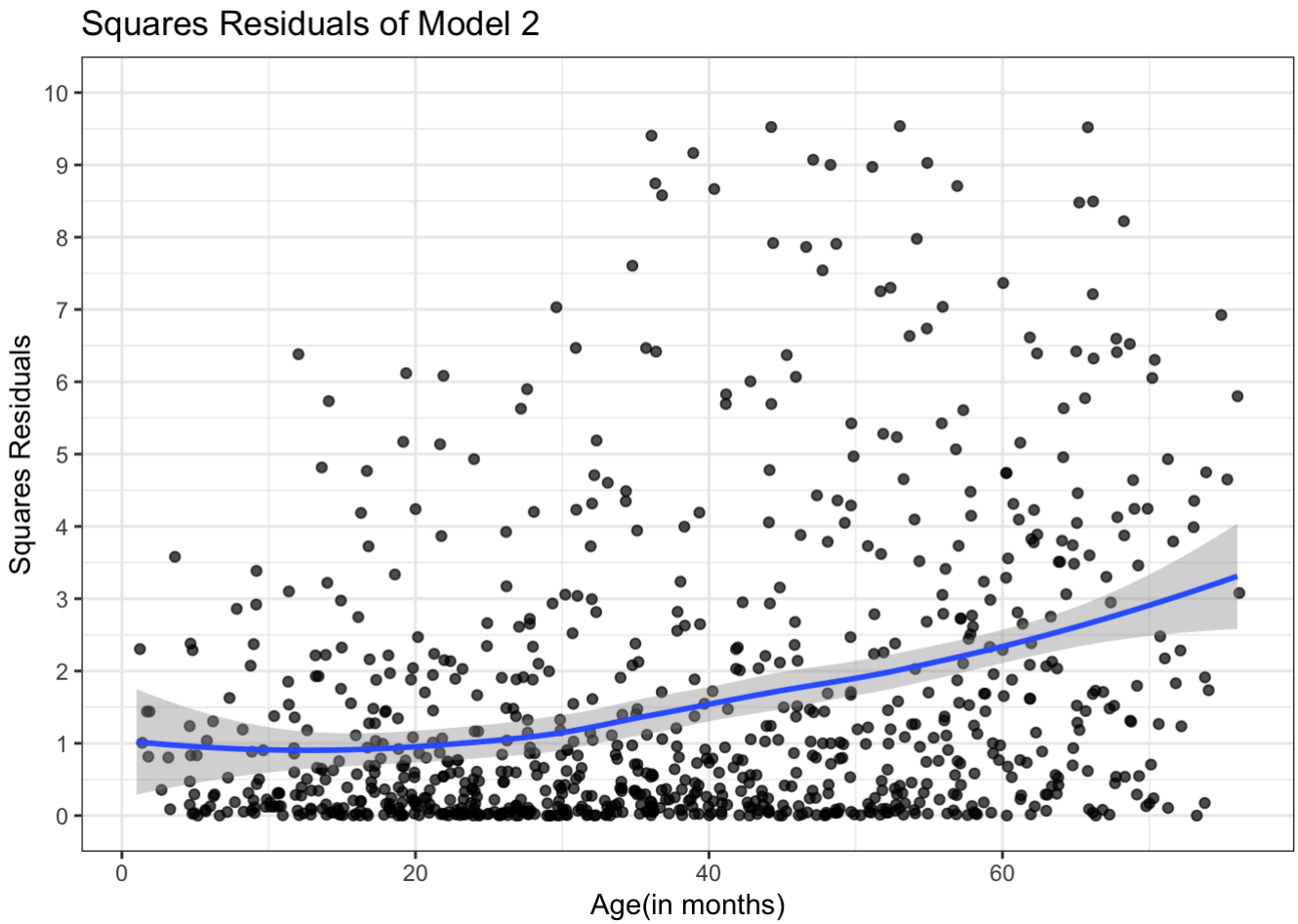
```
##
## Call:
## lm(formula = wt ~ ns(age, df = 5) + parity_num + parity_num:ns(age,
##     df = 5), data = d.cc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9881 -0.8889  0.0698  0.8528  4.6989
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   5.10357    0.36642  13.928   <2e-16 ***
## ns(age, df = 5)1              5.33046    0.37587  14.182   <2e-16 ***
## ns(age, df = 5)2              6.81080    0.50725  13.427   <2e-16 ***
## ns(age, df = 5)3              8.07242    0.38648  20.887   <2e-16 ***
## ns(age, df = 5)4             12.57498    0.88230  14.253   <2e-16 ***
## ns(age, df = 5)5              8.46124    0.46414  18.230   <2e-16 ***
## parity_num                    0.21419    0.64477   0.332    0.740
## ns(age, df = 5)1:parity_num -0.03779    0.64622  -0.058    0.953
## ns(age, df = 5)2:parity_num  0.81273    0.83509   0.973    0.331
## ns(age, df = 5)3:parity_num -0.17174    0.58607  -0.293    0.770
## ns(age, df = 5)4:parity_num -0.16068    1.51735  -0.106    0.916
## ns(age, df = 5)5:parity_num  0.23108    0.65180   0.355    0.723
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.377 on 865 degrees of freedom
## Multiple R-squared:  0.7838, Adjusted R-squared:  0.7811
## F-statistic: 285.1 on 11 and 865 DF,  p-value: < 2.2e-16
```

```
d.cc$fitval2 = model2$fitted.values
d.cc$res2 = model2$residuals
d.cc$r2_2 = d.cc$res2^2
p5 <- ggplot(d.cc,aes(x=age,y=r2_2))+ geom_jitter(alpha=0.7)+ theme_bw()+ geom_smooth
()+ ggtitle("Squares Residuals of Model 2")+
  scale_y_continuous(breaks=seq(0,10.0,1.0),limits=c(0,10.0)) +
  labs(y="Squares Residuals",x="Age(in months)")
p5
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

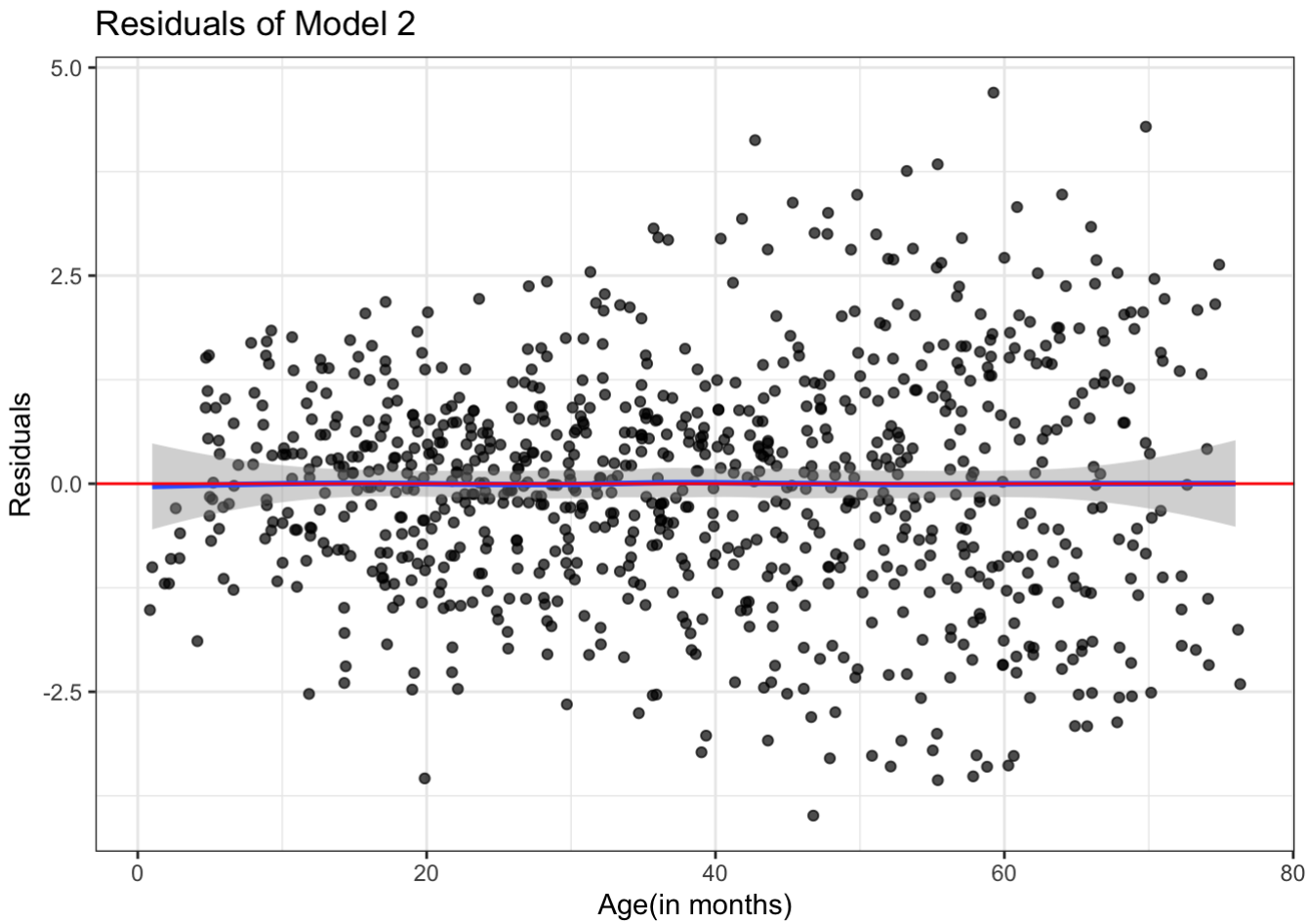
```
## Warning: Removed 24 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```




```
p4 <- ggplot(d.cc,aes(x=age,y=res2))+ geom_jitter(alpha=0.7)+ theme_bw()+ geom_smooth
()+ ggtitle("Residuals of Model 2")+
  geom_hline(yintercept=0,color="red") +
  labs(y="Residuals",x="Age(in months)")
p4
```

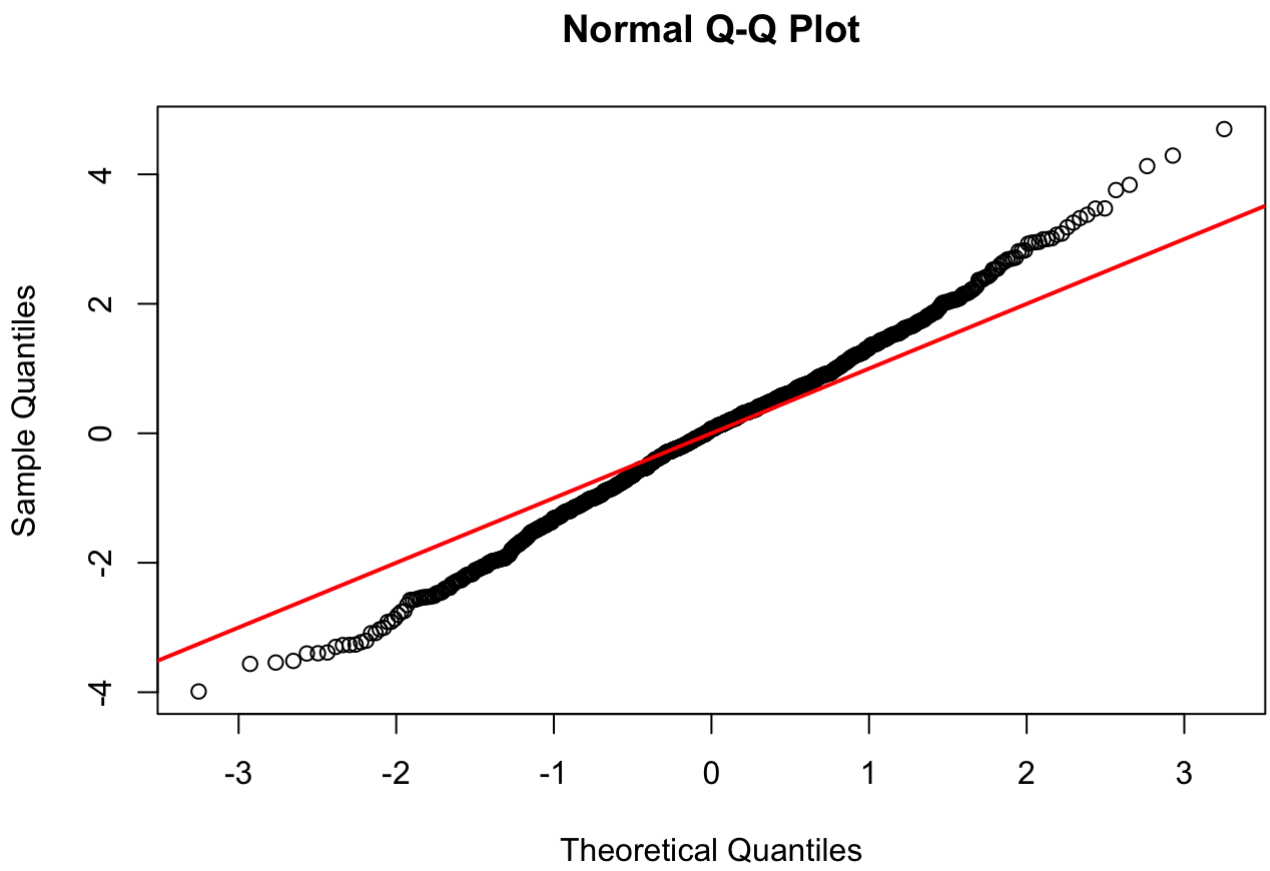
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



After adding the age spline terms (df=5) and the interaction terms of age spline terms(df=5) and parity, the residuals seems to be constant with a mean of zero.

Check for normality of residuals:

```
qqnorm(d.cc$res2)
abline(0,1, col='red', lwd=2)
```

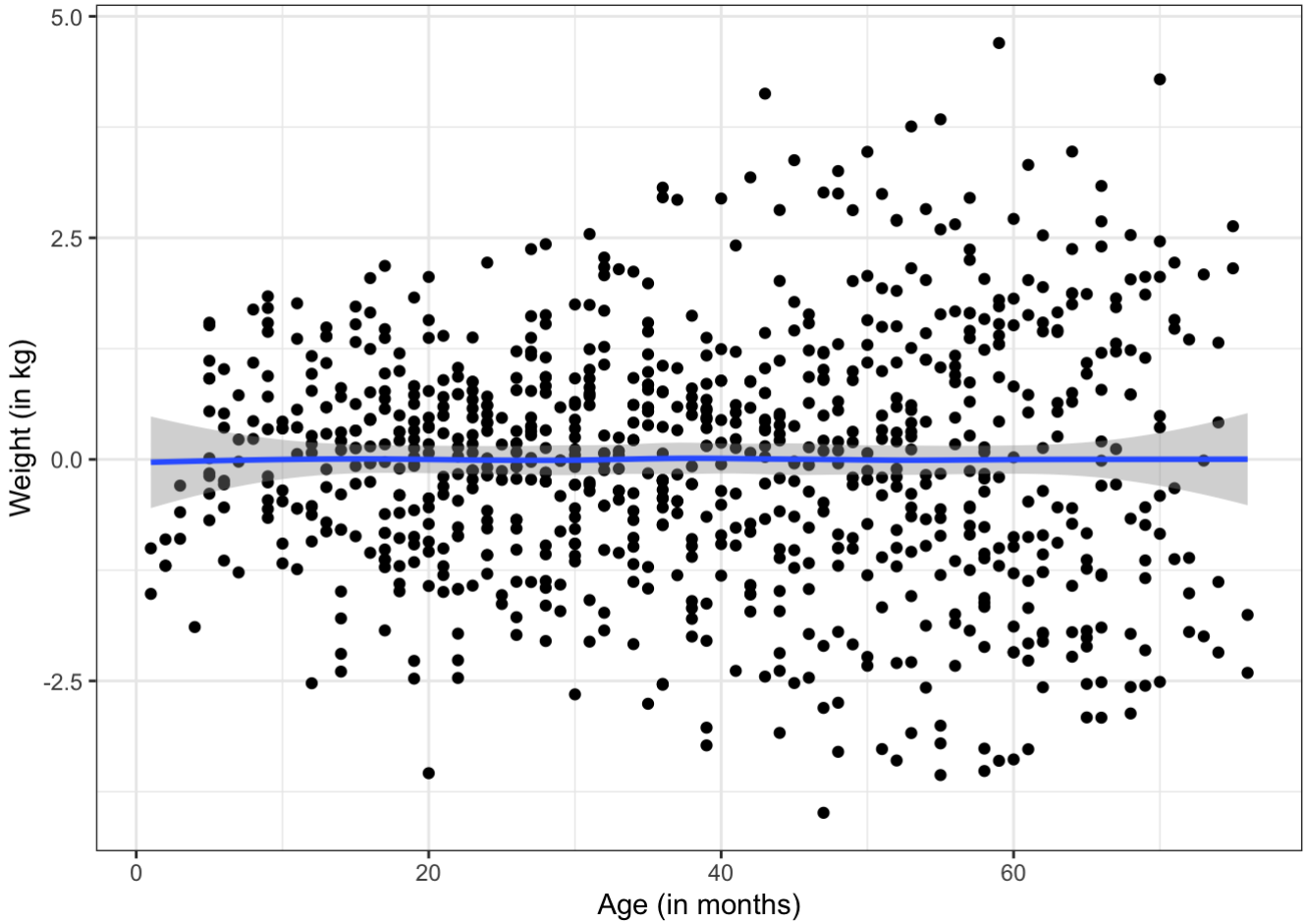


The residuals distribution of model 2 are more ‘normal’ as compared to the distribution of residuals distribution of model 1.

Check for independence of residuals:

```
ggplot(data = d.cc, aes(x = age, y = res2)) + geom_point() + theme_bw() + geom_smooth
() + labs(y="Weight (in kg)",x="Age (in months)")  #scale_y_continuous(breaks=seq(2,1
4,2),limits=c(1.5,14.5)) + #scale_x_continuous(breaks=seq(0,24,6),limits=c(0,24))
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The independent residual variance assumption does not appear to be greatly violated.

```
#perform the likelihood ratio test on two models
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
lrtest(model1,model2)
```

```
## Likelihood ratio test
##
## Model 1: wt ~ age + parity_num + parity_num * age
## Model 2: wt ~ ns(age, df = 5) + parity_num + parity_num:ns(age, df = 5)
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1   5 -1540.0
## 2  13 -1518.8  8 42.566   1.06e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on the p-value(< 0.05) of the likelihood ratio test , and the check for independence of residuals, normality of residuals we would say the proposed new model (model2) is superior to the original model.

Part III: Marginal model for longitudinal data

Use the gls function in R to fit the model you proposed in Part I. From the fit of the model, compute the estimated $Corr(\epsilon_{i0}, \epsilon_{ij})$ for $j = 1, 2, 3, 4$ where the follow-up visits (fuvisit) have values 0 (baseline) and 1,2,3,4.
and use exchangeable variance model and adjust for heteroscedasticity within each age-parity group. According to variance structure check, the variance seems to be quite similar, so we choose to use the exchangeable variance model.

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
```

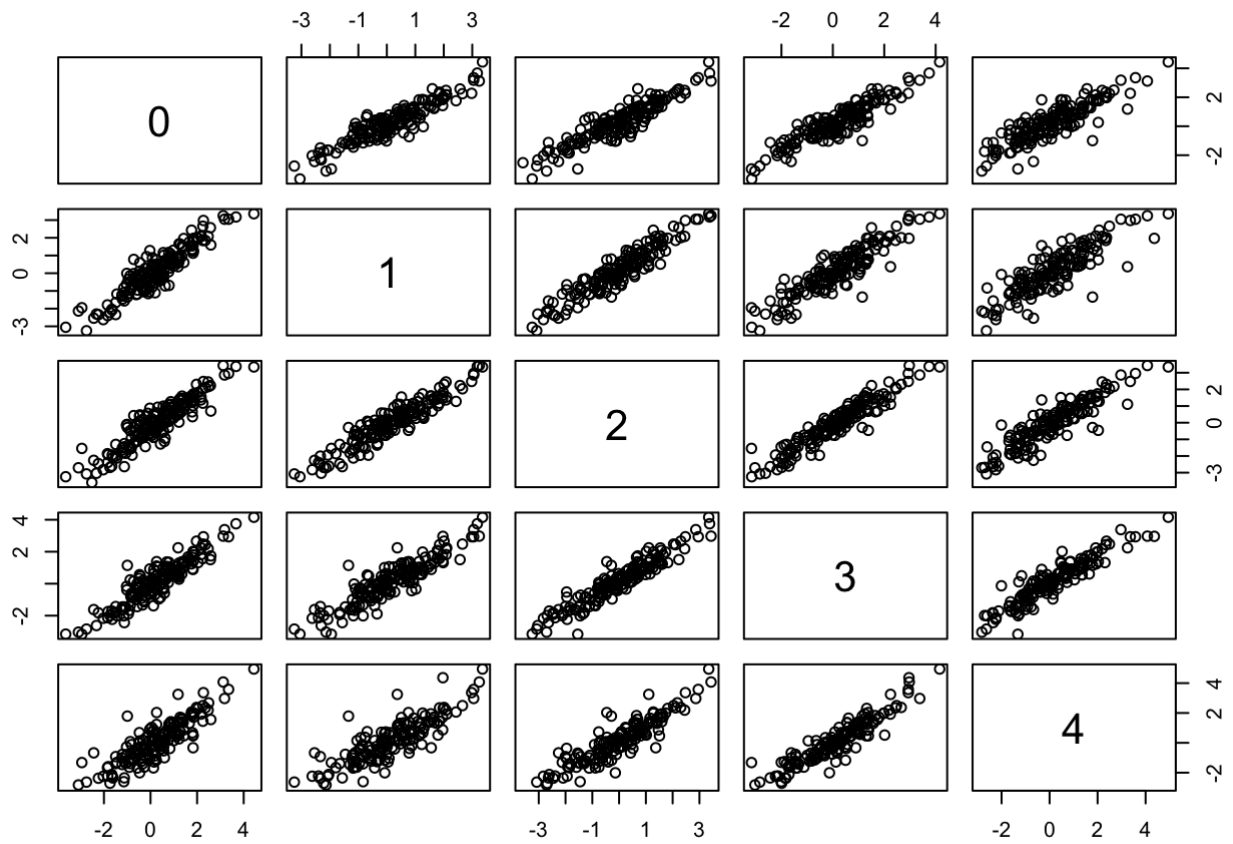
```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
model3 = gls(wt ~ ns(age, df = 5)+ parity_num+parity_num:ns(age, df = 5),
             data= d.cc,
             correlation = corCompSymm(form = ~1|id),
             weights = varFunc(~age+parity_num))
summary(model3)
```

```
## Generalized least squares fit by REML
##   Model: wt ~ ns(age, df = 5) + parity_num + parity_num:ns(age, df = 5)
##   Data: d.cc
##        AIC     BIC    logLik
##   2135.541 2202.22 -1053.771
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | id
##  Parameter estimate(s):
##       Rho
## 0.8372943
## Variance function:
##  Structure: fixed weights
##  Formula: ~age + parity_num
##
## Coefficients:
##                               Value Std.Error  t-value p-value
## (Intercept)                4.566813 0.0704945 64.78254  0.0000
## ns(age, df = 5)1           5.312233 0.1636392 32.46308  0.0000
## ns(age, df = 5)2           7.587929 0.2428962 31.23939  0.0000
## ns(age, df = 5)3           7.506040 0.3010915 24.92944  0.0000
## ns(age, df = 5)4          13.802807 0.3537307 39.02066  0.0000
## ns(age, df = 5)5           8.792896 0.4121326 21.33512  0.0000
## parity_num                 0.251057 0.1347293  1.86342  0.0627
## ns(age, df = 5)1:parity_num  0.394980 0.2653473  1.48854  0.1370
## ns(age, df = 5)2:parity_num  0.158475 0.3658317  0.43319  0.6650
## ns(age, df = 5)3:parity_num  0.248438 0.4375822  0.56775  0.5704
## ns(age, df = 5)4:parity_num -0.013808 0.5345830 -0.02583  0.9794
## ns(age, df = 5)5:parity_num  0.278546 0.5739339  0.48533  0.6276
##
##  Correlation:
##                            (Intr) ns(,d=5)1 ns(,d=5)2 ns(,d=5)3 ns(,d=5)4
## ns(age, df = 5)1           -0.098
## ns(age, df = 5)2           -0.211  0.122
## ns(age, df = 5)3           -0.075  0.285     0.002
## ns(age, df = 5)4           -0.244  0.284     0.434     0.438
## ns(age, df = 5)5           -0.052  0.077     0.311     0.185     0.775
## parity_num                 -0.523  0.051     0.110     0.039     0.127
## ns(age, df = 5)1:parity_num  0.061 -0.617    -0.075    -0.176    -0.175
## ns(age, df = 5)2:parity_num  0.140 -0.081    -0.664    -0.001    -0.288
## ns(age, df = 5)3:parity_num  0.052 -0.196    -0.002    -0.688    -0.301
## ns(age, df = 5)4:parity_num  0.161 -0.188    -0.287    -0.290    -0.662
## ns(age, df = 5)5:parity_num  0.037 -0.056    -0.223    -0.133    -0.557
##                            ns(,d=5)5 prty_n n(,d=5)1: n(,d=5)2: n(,d=5)3:
## ns(age, df = 5)1
## ns(age, df = 5)2
## ns(age, df = 5)3
## ns(age, df = 5)4
## ns(age, df = 5)5
## parity_num                  0.027
## ns(age, df = 5)1:parity_num -0.048    -0.172
## ns(age, df = 5)2:parity_num -0.206    -0.259  0.182
## ns(age, df = 5)3:parity_num -0.127    -0.124  0.319     0.033
## ns(age, df = 5)4:parity_num -0.513    -0.307  0.322     0.484     0.441
## ns(age, df = 5)5:parity_num -0.718    -0.077  0.116     0.318     0.234
```

```
##                                n(,d=5)4:
## ns(age, df = 5)1
## ns(age, df = 5)2
## ns(age, df = 5)3
## ns(age, df = 5)4
## ns(age, df = 5)5
## parity_num
## ns(age, df = 5)1:parity_num
## ns(age, df = 5)2:parity_num
## ns(age, df = 5)3:parity_num
## ns(age, df = 5)4:parity_num
## ns(age, df = 5)5:parity_num  0.716
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -3.2773715 -0.5014200  0.1026855  0.7117432  3.3355439
##
## Residual standard error: 0.2449683
## Degrees of freedom: 877 total; 865 residual
```

```
d.cc$res3 = as.numeric(model3$residuals)
d.cc$r2_3 = (d.cc$res3)^2
## correlation plot
d_wide_3 = d.cc %>% select(id, fuvisit, res3) %>% spread(fuvisit,res3)
pairs(d_wide_3[,-1])
```




```
# the correlation matrix
cor(d_wide_3[,-1], use = 'pairwise.complete.obs')
```

```
##           0         1         2         3         4
## 0 1.0000000 0.9226994 0.9164836 0.9163733 0.8571890
## 1 0.9226994 1.0000000 0.9391856 0.9053748 0.8432054
## 2 0.9164836 0.9391856 1.0000000 0.9450442 0.9008811
## 3 0.9163733 0.9053748 0.9450442 1.0000000 0.9275002
## 4 0.8571890 0.8432054 0.9008811 0.9275002 1.0000000
```

2.Conduct a likelihood ratio test to address the first goal of the analysis; i.e. to determine if the average growth rates of children differ by mother's parity (number of previous live births).

```
model4 = gls(wt ~ ns(age, df = 5)+ parity_num,
             data= d.cc,
             correlation = corCompSymm(form = ~1|id),
             weights = varFunc(~age+parity_num))
summary(model4)
```

```
## Generalized least squares fit by REML
##   Model: wt ~ ns(age, df = 5) + parity_num
##   Data: d.cc
##        AIC      BIC    logLik
##   2128.687 2171.603 -1055.343
##
## Correlation Structure: Compound symmetry
##  Formula: ~1 | id
##  Parameter estimate(s):
##       Rho
## 0.8364945
## Variance function:
##  Structure: fixed weights
##  Formula: ~age + parity_num
##
## Coefficients:
##                     Value  Std.Error  t-value p-value
## (Intercept)       4.572497 0.06889814 66.36604  0.0000
## ns(age, df = 5)1  5.459223 0.12859216 42.45378  0.0000
## ns(age, df = 5)2  7.660344 0.18065024 42.40429  0.0000
## ns(age, df = 5)3  7.593373 0.21759454 34.89689  0.0000
## ns(age, df = 5)4 13.815787 0.26112144 52.90943  0.0000
## ns(age, df = 5)5  8.895810 0.28545789 31.16330  0.0000
## parity_num        0.246307 0.12392875  1.98749  0.0472
##
##  Correlation:
##                  (Intr) n(,d=5)1 n(,d=5)2 n(,d=5)3 n(,d=5)4 n(,d=5)5
## ns(age, df = 5)1 -0.083
## ns(age, df = 5)2 -0.176  0.165
## ns(age, df = 5)3 -0.056  0.309    0.027
## ns(age, df = 5)4 -0.212  0.315    0.470    0.450
## ns(age, df = 5)5 -0.038  0.107    0.319    0.230    0.734
## parity_num       -0.492 -0.069   -0.061   -0.066   -0.069   -0.038
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -3.2909224 -0.5472842  0.1160012  0.7226362  3.3600967
##
## Residual standard error: 0.2443946
## Degrees of freedom: 877 total; 870 residual
```

```
lrtest(model3, model4)
```

```
## Likelihood ratio test
##
## Model 1: wt ~ ns(age, df = 5) + parity_num + parity_num:ns(age, df = 5)
## Model 2: wt ~ ns(age, df = 5) + parity_num
##   #Df  LogLik Df  Chisq Pr(>Chisq)
## 1  14 -1053.8
## 2   9 -1055.3 -5 3.1452     0.6776
```

Model3:

$$Y_{ij} = \beta_0 + \beta_2 ns(age_{ij}, 5) + \beta_2 I(parity_i > 3) + \beta_3 I(parity_i > 3) ns(age_{ij}, 5) + \epsilon_{ij}$$

Model4:

$$Y_{ij} = \beta_0 + \beta_2 ns(age_{ij}, 5) + \beta_2 I(parity_i > 3) + \epsilon_{ij}$$

In output of the likelihood ratio test above, model3 is denoted as 'Model 1' while model 4 is denoted as 'Model1'. With the p-value of 0.68, we fail to reject the null hypothesis that all the added βs are not needed. There is no statistically significant evidence to show that the average growth rates of children differ by mothers' parity at $\alpha = 0.05$ level.

3. you allow the correlation structure to be "independence".The gee function will produce standard error estimates assuming the independence assumption (labeled as "naïve" or "model-based" standard error estimates) and "robust" standard error estimates (using the Huber-White sandwich estimator). Compare the estimated coefficients and standard errors from the gls and gee model fits.

```
library(gee)
```

```
## Warning: package 'gee' was built under R version 4.0.5
```

```
fit = gee(wt~ns(age,5) * parity, data=d.cc,id = id, corstr="independence")
```

```
## Beginning Cgee S-function, @(#) geeformula.q 4.13 98/01/27
```

```
## running glm to get initial regression estimate
```

```
##                  (Intercept)                  ns(age, 5)1
##                   5.10357426                   5.33046190
##                  ns(age, 5)2                  ns(age, 5)3
##                   6.81080018                   8.07241942
##                  ns(age, 5)4                  ns(age, 5)5
##                  12.57498066                   8.46123651
##             parityparity > 3 ns(age, 5)1:parityparity > 3
##                   0.21419315                  -0.03779397
## ns(age, 5)2:parityparity > 3 ns(age, 5)3:parityparity > 3
##                   0.81273412                  -0.17174236
## ns(age, 5)4:parityparity > 3 ns(age, 5)5:parityparity > 3
##                  -0.16067520                   0.23107562
```

```
summary(fit)$coefficients
```

```
##                                Estimate Naive S.E.    Naive z Robust S.E.
## (Intercept)                  5.10357426  0.3664218 13.9281404   0.2635011
## ns(age, 5)1                  5.33046190  0.3758668 14.1817843   0.2788349
## ns(age, 5)2                  6.81080018  0.5072463 13.4270074   0.4682818
## ns(age, 5)3                  8.07241942  0.3864754 20.8872771   0.4269440
## ns(age, 5)4                 12.57498066  0.8822984 14.2525256   0.7160352
## ns(age, 5)5                  8.46123651  0.4641436 18.2297834   0.6282022
## parityparity > 3             0.21419315  0.6447665  0.3322027   0.4725355
## ns(age, 5)1:parityparity > 3 -0.03779397  0.6462230 -0.0584844   0.4728762
## ns(age, 5)2:parityparity > 3  0.81273412  0.8350890  0.9732305   0.7307531
## ns(age, 5)3:parityparity > 3 -0.17174236  0.5860695 -0.2930409   0.6230104
## ns(age, 5)4:parityparity > 3 -0.16067520  1.5173540 -0.1058917   1.1846017
## ns(age, 5)5:parityparity > 3  0.23107562  0.6517997  0.3545194   0.8129966
##                                Robust z
## (Intercept)                  19.36831927
## ns(age, 5)1                  19.11691006
## ns(age, 5)2                  14.54423432
## ns(age, 5)3                  18.90744490
## ns(age, 5)4                  17.56195789
## ns(age, 5)5                  13.46897087
## parityparity > 3              0.45328480
## ns(age, 5)1:parityparity > 3 -0.07992359
## ns(age, 5)2:parityparity > 3  1.11218700
## ns(age, 5)3:parityparity > 3 -0.27566531
## ns(age, 5)4:parityparity > 3 -0.13563648
## ns(age, 5)5:parityparity > 3  0.28422702
```

```
gee_coef = fit$coefficients
gee_se = sqrt(diag(fit$naive.variance))
robust_se = sqrt(diag(fit$robust.variance))
```

```
sqrt(diag(fit$naive.variance))
```

```
##                   (Intercept)                   ns(age, 5)1
##                     0.3664218                     0.3758668
##                   ns(age, 5)2                   ns(age, 5)3
##                     0.5072463                     0.3864754
##                   ns(age, 5)4                   ns(age, 5)5
##                     0.8822984                     0.4641436
##              parityparity > 3 ns(age, 5)1:parityparity > 3
##                     0.6447665                     0.6462230
## ns(age, 5)2:parityparity > 3 ns(age, 5)3:parityparity > 3
##                     0.8350890                     0.5860695
## ns(age, 5)4:parityparity > 3 ns(age, 5)5:parityparity > 3
##                     1.5173540                     0.6517997
```

```
sqrt(diag(fit$robust.variance))
```

```
##                   (Intercept)                     ns(age, 5)1
##                     0.2635011                       0.2788349
##                   ns(age, 5)2                     ns(age, 5)3
##                     0.4682818                       0.4269440
##                   ns(age, 5)4                     ns(age, 5)5
##                     0.7160352                       0.6282022
##             parityparity > 3 ns(age, 5)1:parityparity > 3
##                     0.4725355                       0.4728762
## ns(age, 5)2:parityparity > 3 ns(age, 5)3:parityparity > 3
##                     0.7307531                       0.6230104
## ns(age, 5)4:parityparity > 3 ns(age, 5)5:parityparity > 3
##                     1.1846017                       0.8129966
```

```
library(knitr)
gls_coef = model3$coefficients
gls_se = sqrt(diag(model3$varBeta))
compare <- cbind(round(gee_coef,3),round(gee_se,3), round(robust_se,3),round(gls_coe
f,3), round(gls_se,3))
compare <- as.matrix(compare)
colnames(compare) = c("gee_coef","gee_se","robust_se","gls_coef","gls_se")
knitr::kable(compare, col.names = gsub("[_]", " ", colnames(compare)),align = "lccrr"
)
```

	gee coef	gee se	robust se	gls coef	gls se
(Intercept)	5.104	0.366	0.264	4.567	0.070
ns(age, 5)1	5.330	0.376	0.279	5.312	0.164
ns(age, 5)2	6.811	0.507	0.468	7.588	0.243
ns(age, 5)3	8.072	0.386	0.427	7.506	0.301
ns(age, 5)4	12.575	0.882	0.716	13.803	0.354
ns(age, 5)5	8.461	0.464	0.628	8.793	0.412
parityparity > 3	0.214	0.645	0.473	0.251	0.135
ns(age, 5)1:parityparity > 3	-0.038	0.646	0.473	0.395	0.265
ns(age, 5)2:parityparity > 3	0.813	0.835	0.731	0.158	0.366
ns(age, 5)3:parityparity > 3	-0.172	0.586	0.623	0.248	0.438
ns(age, 5)4:parityparity > 3	-0.161	1.517	1.185	-0.014	0.535
ns(age, 5)5:parityparity > 3	0.231	0.652	0.813	0.279	0.574

The coefficients estimate from `gee` is similar as compared to those from `gls` . For the non-interaction terms, sometimes `gee` coefficients are larger while sometimes `gls` coefficients are larger. In most cases, the `gls` standard error are smaller than the `gee` unstructured standard error estimate and robust standard error estimate. The `gee` unstructured standard error estimate and robust standard error estimate are similar and do not have an obvious pattern.

For the interaction terms, the coefficients estimate from `gee` are sometimes in different directions as compared to those from `gls` . This may because as we have proved through likelihood ratio test that there is no evidence that the interaction terms are needed, so these coefficients may not be meaningful and are fairly close to 0. The `gls` standard error for the interaction terms are smaller than the `gee` unstructured standard error estimate and robust standard error estimate. The `gee` unstructured standard error estimate and robust standard error estimate are similar and do not have an obvious pattern.

4.The bootstrap procedure can also be applied to longitudinal or clustered data to estimate standard errors of estimated coefficients (or functions of). To preserve the within-subject dependency, the bootstrap procedure samples children (with replacement) as opposed to assessments. See the `ProblemSet3.rmd` file for code to implement a clustered bootstrap. Compute the bootstrap standard error estimates and compare these to the standard errors from the `gls` and `gee` model fits. Comment on similarities and differences.

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
```

```
## ✓ tibble  3.1.6     ✓ stringr 1.4.0
## ✓ readr   2.1.1     ✓ forcats 0.5.1
## ✓ purrr   0.3.4
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x nlme::collapse() masks dplyr::collapse()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(gridExtra)
```

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

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

```
library(boot)
setwd("~/Documents/JHU/term3/biostatmethods3/Biostat653")
load("nepal.anthro.rdata")
d1 = nepal.anthro[,c("id", "alive", "age", "wt","fuvisit")]
d1$parity = factor(ifelse(d1$alive <= 4, 0, 1),
            levels = 0 : 1,
            labels = c("1 to 4 live births",
                       "5 or more live births"))
# Keep only assessments with no missing age, wt or parity
d2 = d1[complete.cases(d1),]
```

Longitudinal or clustered data bootstrap procedure

Create a function that will take a bootstrap sample of children (with replacement) and fit the mean model of interest.
The bootstrap procedure will require some transformations of the data from long to wide to long again.

```
# Create a wide version of the data
# Each row represents an individual child
nepal.wide <- d2[,c('id','parity','age','wt','fuvisit')] %>% pivot_wider(id_cols=c(i
d,parity),values_from = c(age,wt),names_from='fuvisit')
set.seed(523)
## Write a bootstrap function
my.boot <- function(data, id){
  # Resample the children
  dt <- data[id, ]
  # Create a new id variable and drop the old id
  dt$id = NULL
  dt$id = seq(1,nrow(dt))
  # Convert to the long format for model fitting
  dlong0 = pivot_longer(dt,cols=!c(id,parity),
                   names_to=c("vars","fuvisit"),
                   names_sep="_",values_to = "y")
  dlong = pivot_wider(dlong0,names_from="vars",values_from="y")
  # Fit the mean model
  # NOTE:  We can use a ordinary least squares procedure here
  # since this procedure produces unbiased estimates of the model
  # coefficients even when the correlation or variance assumption
  # is violated
  fit = lm(wt ~ ns(age, 5) * parity, dlong)
  coefficients(fit)
}

result = boot(nepal.wide, my.boot, 1000)
boot.se <- apply(result$t,2,FUN=function(x) sqrt(var(x)))
boot.se <- round(boot.se,3)
compare_2 = cbind(compare,boot.se)
colnames(compare_2) = c("gee_coef","gee_se","robust_se","gls_coef","gls_se","boot_se"
)
knitr::kable(compare_2, col.names = gsub("[_]", " ", colnames(compare_2)),align = "lc
crr")
```

	gee coef	gee se	robust se	gls coef	gls se	boot se
(Intercept)	5.104	0.366	0.264	4.567	0.070	0.288
ns(age, 5)1	5.330	0.376	0.279	5.312	0.164	0.402
ns(age, 5)2	6.811	0.507	0.468	7.588	0.243	0.650
ns(age, 5)3	8.072	0.386	0.427	7.506	0.301	0.566
ns(age, 5)4	12.575	0.882	0.716	13.803	0.354	0.862
ns(age, 5)5	8.461	0.464	0.628	8.793	0.412	0.914

	gee coef	gee se	robust se	gls coef	gls se	boot se
parityparity > 3	0.214	0.645	0.473	0.251	0.135	0.510
ns(age, 5)1:parityparity > 3	-0.038	0.646	0.473	0.395	0.265	0.638
ns(age, 5)2:parityparity > 3	0.813	0.835	0.731	0.158	0.366	0.792
ns(age, 5)3:parityparity > 3	-0.172	0.586	0.623	0.248	0.438	0.843
ns(age, 5)4:parityparity > 3	-0.161	1.517	1.185	-0.014	0.535	1.208
ns(age, 5)5:parityparity > 3	0.231	0.652	0.813	0.279	0.574	1.129

It is obvious that the bootstrap standard error is greater than both the `gls` standard error but similar to the `gee` unstructured standard error estimate and the robust standard error estimate.
Take the standard error of the coefficient of 'parity' as an example, from the unstructured estimate it's 0.645, from bootstrap it's 0.510, from the robust variance estimator it's 0.473, from the `gls` methods it's 0.135.

Part IV: Linear mixed model motivation!

we can conduct some simple intuitive analyses that would allow us to explore variation in growth rates. In what follows, we will assume that growth is linear. This is a strong assumption that is likely violated but will keep the analyses simple.

1.Fit a simple linear regression of weight on age (linear) for each child in the sample and save the estimated slope.

```
idlist <- unique(d2$id)
slopes <- NA
baseline_age <- NA
for (i in 1:length(idlist)) {
  dt <- d2 %>% filter(id==idlist[i])
  SLR <- lm(wt~age, data = dt)
  slopes[i] <- coef(SLR)[2]
  baseline_age[i] <- dt %>% filter(fuvisit==0) %>% select(age) %>% as.numeric()
}
head(slopes)
```

```
## [1] 0.08250000 0.11500008 0.12750001 0.08071427 0.05249999 0.12500002
```

```
avg_slope <- mean(slopes, na.rm = TRUE)
avg_slope
```

```
## [1] 0.1335083
```

2.Scale the estimated slopes to represent the expected change in weight per year.

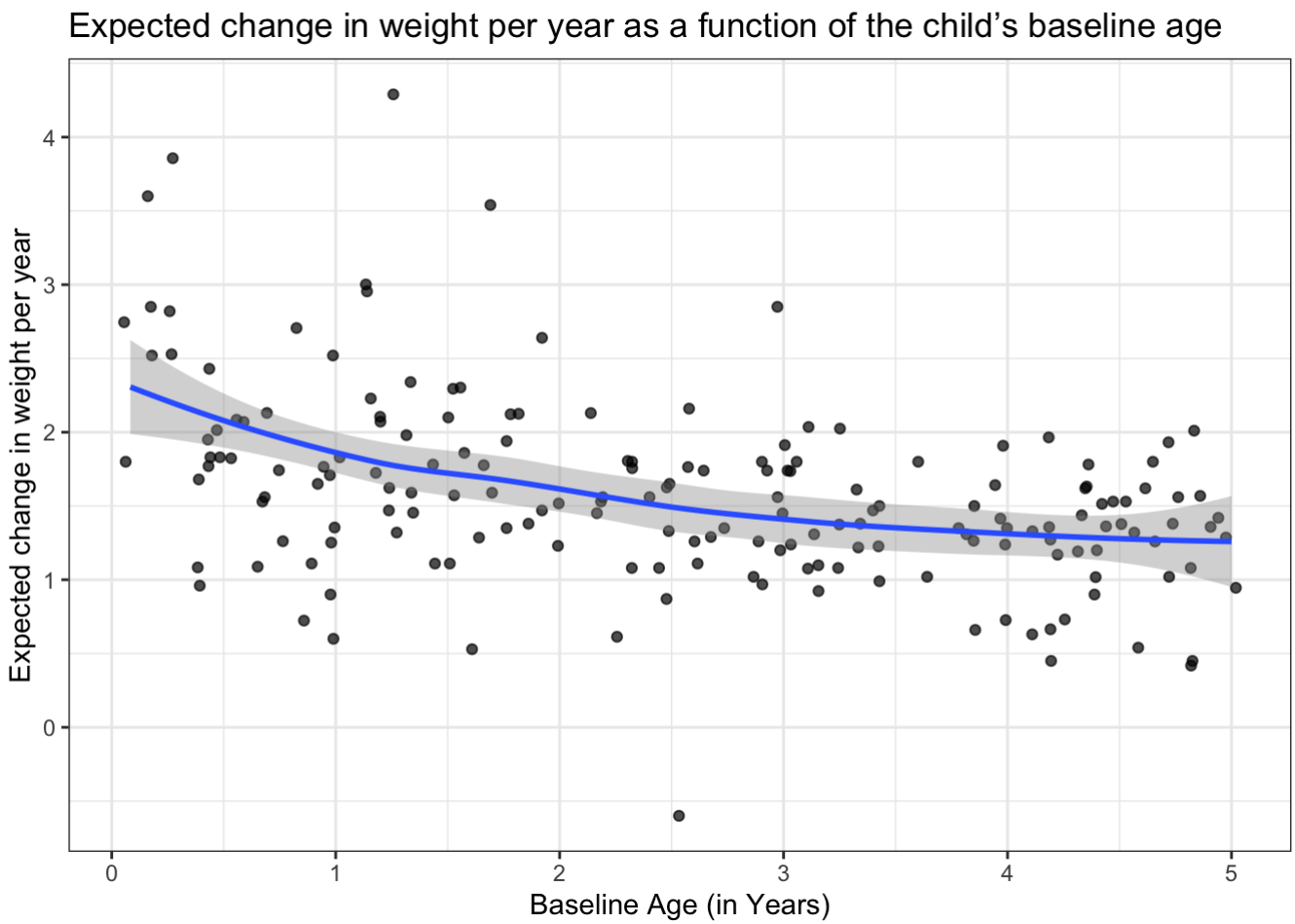
```
sc_slopes <- slopes*12
mean(sc_slopes, na.rm = T) ## Children on average will grow 1.60 kg per year
```

```
## [1] 1.602099
```

3.Plot the expected change in weight per year as a function of the child's baseline age (i.e. age when fuvisit = 0). Describe any patterns you observe in:

```
dat3 <- data.frame(idlist, sc_slopes, baseline_age)
dat3.cc <- dat3[complete.cases(dat3),]
ggplot(aes(x=baseline_age/12,y=sc_slopes), data= dat3.cc)+ geom_jitter(alpha=0.7)+
  theme_bw()+
  geom_smooth()+
  ggtitle("Expected change in weight per year as a function of the child's baseline a
ge")+
  labs(y="Expected change in weight per year",x="Baseline Age (in Years)")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



a.The population average change in weight per year as a function of the child's baseline age

```
fita <- lm(sc_slopes~baseline_age,data=dat3.cc)
summary(fita)
```

```
##
## Call:
## lm(formula = sc_slopes ~ baseline_age, data = dat3.cc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18736 -0.32935 -0.03074  0.29009  2.46661
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.059418   0.086267  23.873  < 2e-16 ***
## baseline_age -0.015735   0.002453  -6.415 1.25e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5758 on 177 degrees of freedom
## Multiple R-squared:  0.1886, Adjusted R-squared:  0.1841
## F-statistic: 41.15 on 1 and 177 DF,  p-value: 1.245e-09
```

The expected change in weight per year decreases as baseline age increase. And if I regress the scaled slope agaist baseline age, I would get that the estimated change in slope is -0.016 per month of age (p=1.25e-09), suggesting statistically significant decrease in slope as age grows.

b.Variation in the expected change in weight per year across children as a function of the child's baseline age (be quantitative, i.e. estimate the variance)

```
dat3b <- d.cc %>% select(id, age, fuvisit, parity_num, parity) %>% filter(fuvisit==0)
colnames(dat3) <- c("id","sc_slopes","baseline_age")
dat3b_cal <- inner_join(dat3b, dat3, by='id')
dat3b_cal <- dat3b_cal[complete.cases(dat3b_cal),]
dat3b_cal <- dat3b_cal %>% mutate(agecat = ifelse(age<=30,0,1))
## put age into 2 categories, 0~3 years(young) and 3~6 years(old)
table(dat3b_cal$agecat)
```

```
##
##  0  1
## 90 89
```

```
dat3b_cal %>% group_by(agecat) %>% summarise(variance = var(sc_slopes, na.rm = T))
```

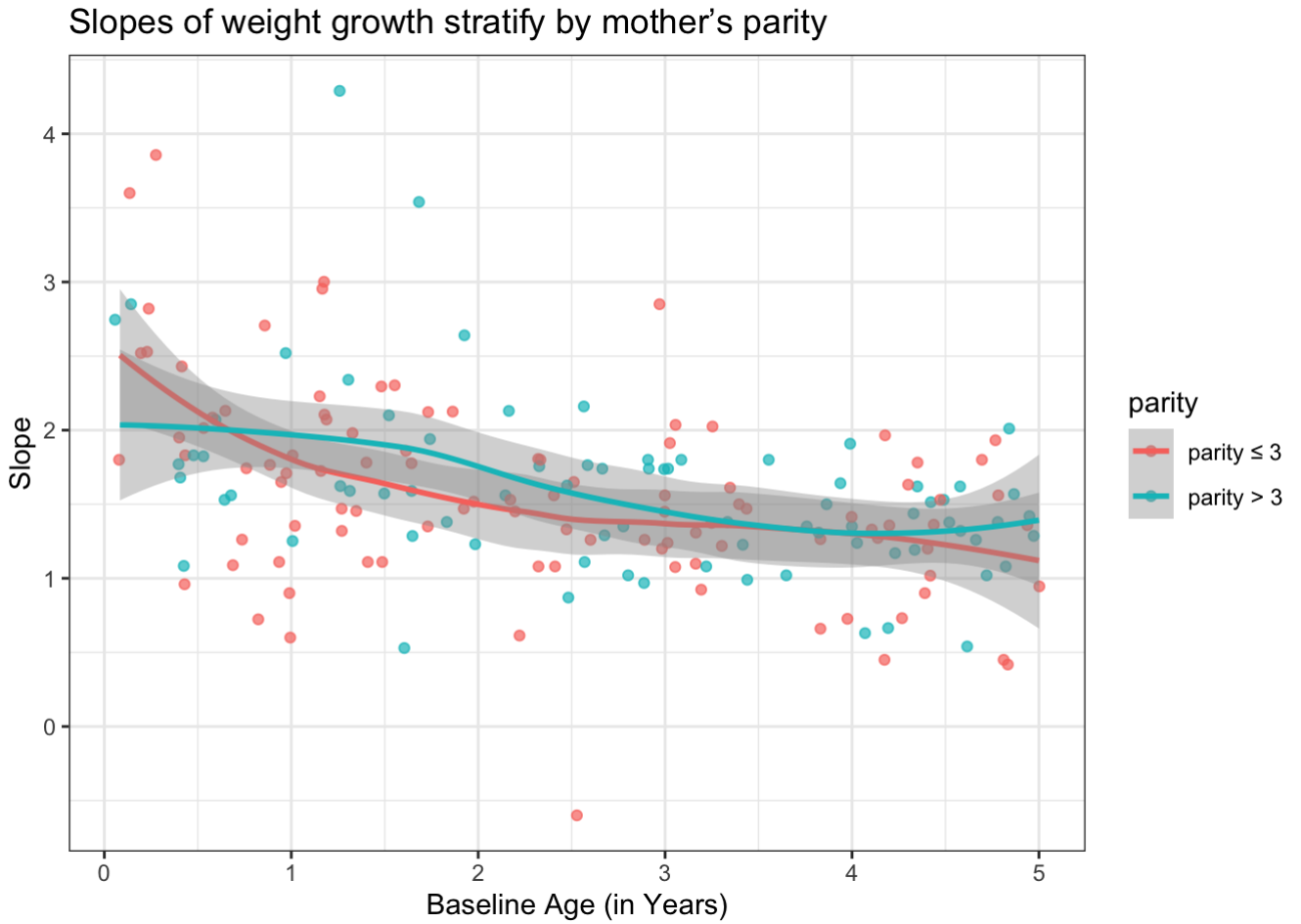
```
## # A tibble: 2 × 2
##   agecat variance
##    <dbl>    <dbl>
## 1      0    0.545
## 2      1    0.173
```

It is clear that children aged 0~3 years(young) has higher variance in expected change rate in weight per year (variance = 0.545) as compared to those who are 3~6 years(old) (variance = 0.173).

4. Repeat 3) but stratify by mother's parity.
a.The population average change in weight per year as a function of the child's baseline age stratify by mother's parity

```
p41 <- ggplot(aes(x=baseline_age/12,y=sc_slopes, color= parity), data= dat3b_cal)+ ge
om_jitter(alpha=0.7)+
  theme_bw()+
  geom_smooth()+
  ggtitle("Slopes of weight growth stratify by mother's parity")+
  labs(y="Slope",x="Baseline Age (in Years)")
p41
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
fita2 <- lm(sc_slopes~baseline_age+parity,data=dat3b_cal)
summary(fita2); round(confint(fita2),3)
```

```
##
## Call:
## lm(formula = sc_slopes ~ baseline_age + parity, data = dat3b_cal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.14548 -0.31789 -0.00681  0.26869  2.40409
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.030642   0.090007  22.561  < 2e-16 ***
## baseline_age     -0.016172   0.002483  -6.514  7.4e-10 ***
## parityparity > 3  0.097854   0.087970   1.112    0.268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5754 on 176 degrees of freedom
## Multiple R-squared:  0.1943, Adjusted R-squared:  0.1851
## F-statistic: 21.22 on 2 and 176 DF,  p-value: 5.538e-09
```

```
##                    2.5 % 97.5 %
## (Intercept)        1.853  2.208
## baseline_age      -0.021 -0.011
## parityparity > 3 -0.076  0.271
```

The expected change in weight per year decreases as baseline age increase. This pattern is consistent across two parity groups.

And if I regress the scaled slope agaist baseline age, I would get that the estimated change in slope is -0.016 (95%CI [-0.021,-0.011]) per month of age for parity lt3 group , suggesting statistically significant decrease in slope as age grows. The estimated change in slope is 0.081er month of age for parity

$$3group. The estimated difference in growth rate between parity$$

3 group ad parity lt3 group is 0.098 (95%CI[-0.076,0.271]), suggesting no statistically significant difference in weight growth rate between the two groups.

b.Variation in the expected change in weight per year across children as a function of the child's baseline age (be quantitative, i.e. estimate the variance)stratify by mother's parity

```
dat3b_cal %>% group_by(parity, agecat) %>% summarise(variance = var(sc_slopes, na.rm
= T))
```

```
## `summarise()` has grouped output by 'parity'. You can override using the `.groups`
argument.
```

```
## # A tibble: 4 × 3
## # Groups:   parity [2]
##   parity     agecat variance
##   <fct>       <dbl>    <dbl>
## 1 parity ≤ 3      0    0.533
## 2 parity ≤ 3      1    0.227
## 3 parity > 3      0    0.577
## 4 parity > 3      1    0.126
```

The variance patterns are consistent with the previous question: The variance of slopes decrease as age grows in both parity groups. In the age category 0 (0-2 years old), parity > 3 group's variance of slope 0.577 and is larger than parity ≤ 3 group which is 0.533. However, for older children, parity > 3 group's slope's variance of slope is 0.227 and is less than parity ≤ 3 group which is 0.126.
Part V: Summarize your findings ##### Write a brief report with sections: objective, data, methods, results, summary as if for a health services journal. You may include up to 2 figures (which may have multiple panels). Remember to be enumerate when possible!

Introduction & exploratory data analysis

In this problem set, I analyze 1000 records on anthropologic data of 197 Nepalese kids at 5 time points, the time between each record t are approximately 4 months. The key variables s of this dataset are id (child's id) , age (age in months), wt (Child's weight measured in kilograms), and parity (the number of kids the mother has ever had born alive).

I categorize woman (mother of the children) into two categories, parity ≤ 3 and parity > 3 by how many live births she had gave birth to before. After exploratory analysis, I find there are 610 records with parity ≤ 3, and 390 records with parity > 3, which is equivalent to 122 children born by mother with parity ≤ 3, and 78 children born by mother with parity > 3. However, some of the visits contain missing data. I remove rows with missing data on parity, weight, age or id, leaving 877 complete records for further analysis.

Methods

1. We then visualized how each children's weight change as age grows by a spaghetti plot.
2. I first fit a linear model (model1) to the data by set responsive variable as weight, covariates as age, parity group indicator variable, and the interaction term of age and parity group (assuming constant variance and no interactions), and we check this model through several assumptions. I plot residuals and squared residuals against age, use Q-Q plot to check normality of residuals, visualize influential variable as well as outliers and variables with high leverage. Correlations Structure is explored by the scatter plot of residuals and the calculated correlation matrix.
3. I proposed a new model (model2) by adding natural splines terms with 5 degrees of freedom to the age variable and iterate the model checking and correlation sturcture exploring procedures in the previous step.
4. Sensitivity analysis: I then fit marginal model using `gls` function (model 3) and calculate the estimated correlations between error terms between the baseline visit and the four follow-up visits. A likelihood ratio test is conducted to see if the average growth rates of children differ by mother's parity.
5. `gee` function is used to fit the model to the data and I calculated standard error estimates assuming the independence assumption and using the Huber-White sandwich estimator. I also estimate the standard error using bootstrap method.
6. Then I fit a simple linear regression of weight on age for each child in the sample and plot the slopes against baseline age. I examine variation in the expected change in weight per year within each one year of age.

Results:

1. The spaghetti shows that Children's weight varies more as age grows older, but growth rates between two parity groups are very similar.
2. Model 1 is not a good fit of the data. Constant variance assumption are violated. Correlation between residuals are close to 1, suggesting violation of the independent error assumptions strong correlations between residuals. There are many outliers as well as many data points with high leverage/influence. The independence assumption on residuals is clearly violated. There are strong correlations between residuals.
3. Model 2 is a better fit of the data. After adding the age spline terms (df=5) and the interaction terms of age spline terms(df=5) and parity, the residuals seem to be constant with a mean of zero. The residuals distribution of model 2 are more 'normal' as compared to the distribution of residuals distribution of model 1. However, the independent error assumptions are still violated. I perform a likelihood ratio test between Model1 and Model 2, and I would reject the null hypothesis with a p-value 0f 1.06e-06. There is significant evidence that Model 2 is a better fit of the data as compared to Model 1.

```
grid.arrange(p2,p3,p4,p5,nrow=2,top = "Assumption checking ")
```

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

```
## Warning: Removed 27 rows containing non-finite values (stat_smooth).
```

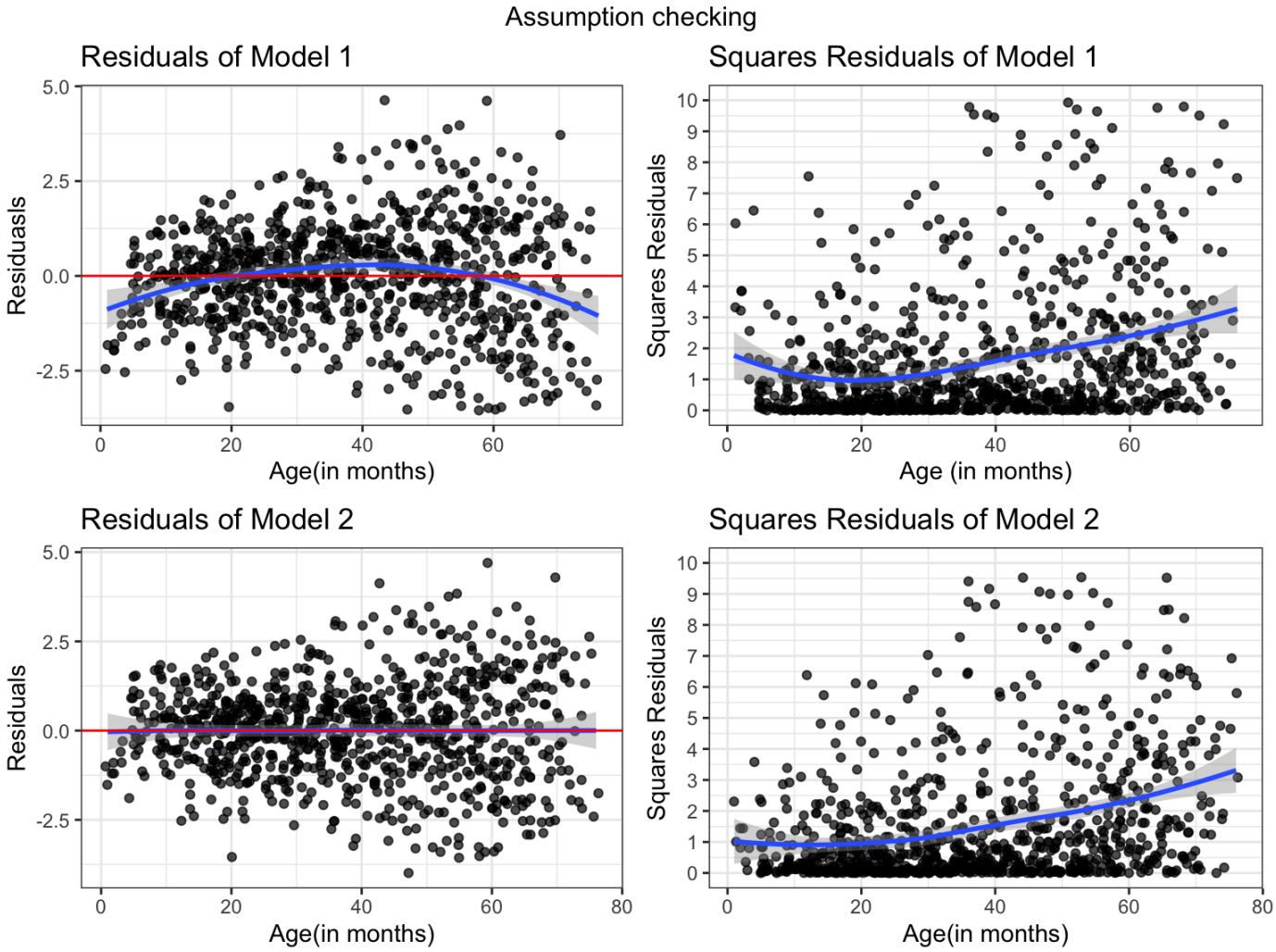
```
## Warning: Removed 27 rows containing missing values (geom_point).
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

```
## Warning: Removed 24 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 24 rows containing missing values (geom_point).
```



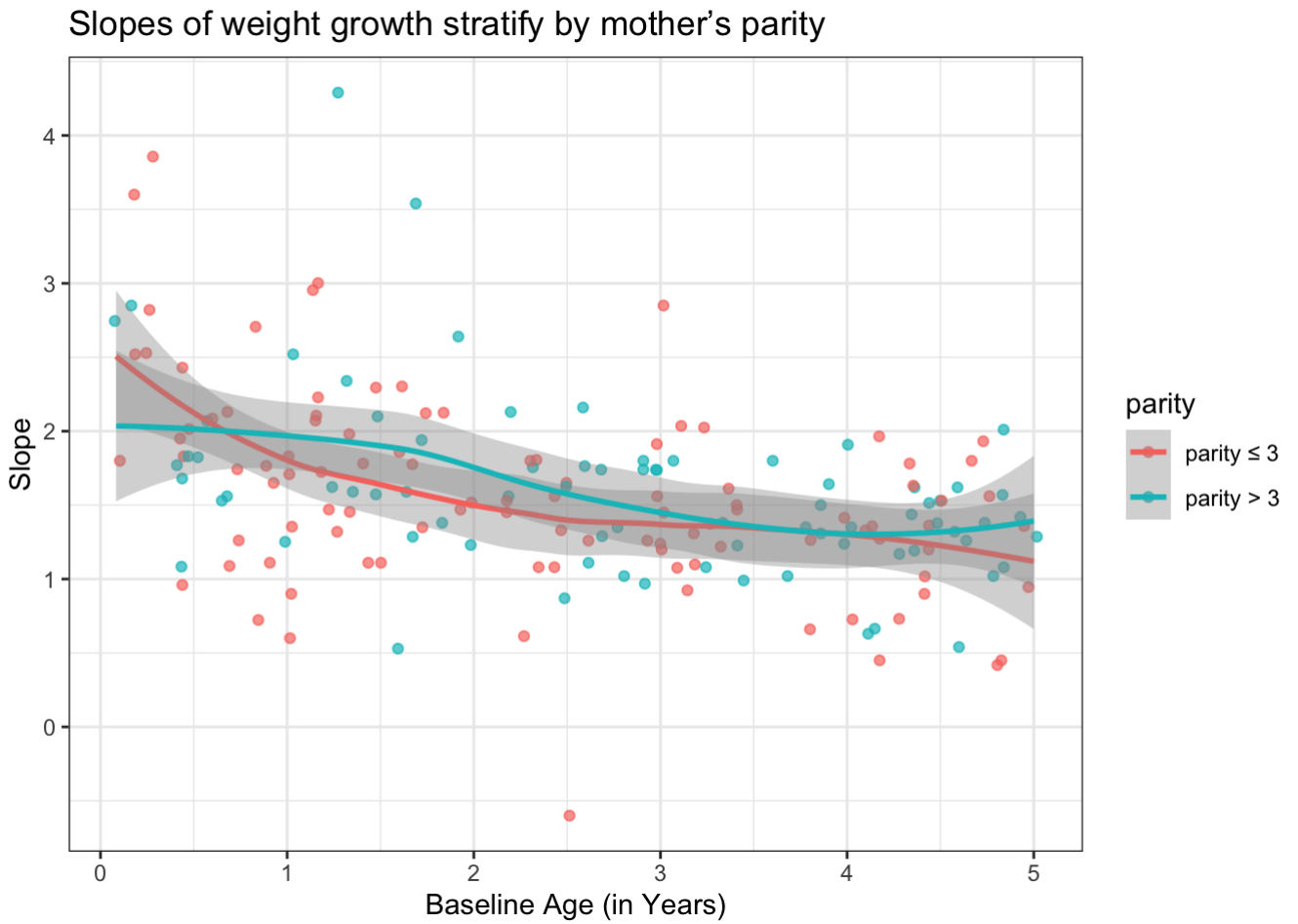
4. The likelihood ratio test between `gls` model with and without the interection terms demonstrate that, with the p-value of 0.68, we fail to reject the null hypothesis that all the added βs are not needed. There is no statistically significant evidence to show that the average growth rates of children differ by mothers' parity at $\alpha = 0.05$ level.

5. The coefficients estimate from `gee` is similar as compared to those from `gls` . The `gee` unstructured standard error estimate and robust standard error estimate are similar and do not have an obvious pattern.
   For the interaction terms, the coefficients estimate from `gee` are sometimes in different directions as compared to those from `gls` . This may because as we have proved through likelihood ratio test that there is no evidence that the interaction terms are needed, so these coefficients may not be meaningful and are fairly close to 0. The `gls` standard error for the interaction terms are smaller than the `gee` unstructured standard error estimate and robust standard error estimate. The `gee` unstructured standard error estimate and robust standard error estimate are similar and do not have an obvious pattern.

The bootstrap standard error is much greater than the `gls` standard error but similar to the `gee` unstructured standard error estimate and the robust standard error estimate. Take the standard error of the coefficient of 'parity' as an example, from the unstructured estimate it's 0.645, from bootstrap it's 0.510, from the robust variance estimator it's 0.473, from the `gls` methods it's 0.135.

```
p41
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



6. Estimate the population variation in annual growth rates: From the plot and by calculating the variances, I observe youngest kids have larger variance in slopes as compared to older kids, and this pattern is consistent across both parity groups. The variance in in expected change rate in weight per year of children aged 0~3 years(young) is 0.545 while it is 0.173 of children aged 3~6 years(old). However, the differece in slope between two parity groups are not statistically significant.
   In the age category 0 (0-3 years old), parity > 3 group's variance of slope 0.64 and is larger than parity ≤ 3 group which is 0.48. However, for older children, parity > 3 group's variance of slope is less than parity ≤ 3 group.

Conclusion / Summary

According to visualizations (stratified by parity groups) and the result of likelihood ratio test (p>0.05), the average growth rates of children is not statistically different across two parity groups.
For population variation in annual growth rates, younger children will have higher variation in annual growth rates, and this pattern in consistent across two parity groups.