

## Problem Set 3 Longitudinal Bootstrap Code

```
knitr::opts_chunk$set(warning = FALSE)
library(nlme)
library(tidyverse)
library(gee)
library(lmtest)
library(splines)
library(ggplot2)
library(gridExtra)
library(dplyr)
library(boot)

options(digits = 3)
```

### Part I: Get familiar with the data

#### Load the data and create the parity variable

1. Read in the Nepal study data, keep only the necessary variables, create the weight-for-age z-scores and indicator for female sex and exclude visits with missing weight-for-age z-scores or where children are over 60 months of age.

```
load("nepal.anthro.rdata")
d0 = nepal.anthro[,c("id", "alive", "age", "wt", "fuvisit", "sex")]
d0$female = factor(ifelse(d0$sex==2,1,0), levels=0:1, labels=c("Male", "Female"))
# install.packages("anthro")
library(anthro)
zscores = with(d0, anthro_zscores(sex = sex, age = age, weight = wt, is_age_in_month=TRUE))$zwei
d = cbind(d0, zscores)[complete.cases(d0) & d0$age<=60,]
d_clean <- d |>
  filter(!is.na(zscores) | zscores != '')
summary(d_clean$zscores)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      -6.12  -2.54   -1.88   -1.97  -1.27    0.71
```

2. Identify the number of children in the sample, overall and for each sex.

```
# number of children in the sample
n_distinct(d_clean$id)      #1: How many distinct participants? 195

## [1] 195

# Number of children by sex
n_child_sex = d_clean %>%
  group_by(female) %>%
  summarise(num_children = n_distinct(id))
```

```
n_child_sex
```

```
## # A tibble: 2 x 2
##   female num_children
##   <fct>      <int>
## 1 Male         102
## 2 Female        93
```

```
# number of visits for each child with non-missing z-scores
visits_per_child <- aggregate(zscores ~ id, data=d_clean, function(x) sum(!is.na(x)))
names(visits_per_child) <- c("id", "num_visits")
visits_per_child
```

Calculate the number of visits for each child with non-missing weight-for-age z-scores and compute the average and quartiles of the number of visits for each child by sex.

```
##      id num_visits
## 1  120011         4
## 2  120012         1
## 3  120021         5
## 4  120022         4
## 5  120023         3
## 6  120031         3
## 7  120051         5
## 8  120052         5
## 9  120053         2
## 10 120061         5
## 11 120062         1
## 12 120071         5
## 13 120072         2
## 14 120081         5
## 15 120082         5
## 16 120091         5
## 17 120111         3
## 18 120112         2
## 19 120121         4
## 20 120131         5
## 21 120132         5
## 22 120133         2
## 23 120141         5
## 24 120142         2
## 25 120151         5
## 26 120152         5
## 27 120161         5
## 28 120162         5
## 29 120191         5
## 30 120192         4
## 31 120211         4
## 32 120212         5
## 33 120231         5
## 34 120241         1
## 35 120242         1
## 36 120251         5
```

## 37	120252	3
## 38	120261	5
## 39	120271	4
## 40	120291	4
## 41	120292	2
## 42	120321	5
## 43	120322	3
## 44	120331	2
## 45	120341	5
## 46	120342	3
## 47	120351	5
## 48	120352	5
## 49	120361	5
## 50	120362	5
## 51	120371	5
## 52	120372	5
## 53	120381	5
## 54	120382	2
## 55	120383	5
## 56	120384	4
## 57	120391	5
## 58	120392	5
## 59	120401	5
## 60	120402	5
## 61	120411	3
## 62	120412	3
## 63	120431	5
## 64	120441	5
## 65	120442	1
## 66	120443	3
## 67	120444	4
## 68	120451	5
## 69	120452	5
## 70	120453	1
## 71	120471	5
## 72	120472	5
## 73	120473	1
## 74	120541	5
## 75	120561	5
## 76	120571	5
## 77	120572	5
## 78	120581	5
## 79	120582	5
## 80	120591	5
## 81	120601	5
## 82	120602	5
## 83	120603	5
## 84	120611	5
## 85	120631	5
## 86	120632	1
## 87	120651	5
## 88	120671	5
## 89	120672	4
## 90	120681	5

## 91	120682	5
## 92	120683	4
## 93	120691	1
## 94	120692	5
## 95	360011	3
## 96	360012	3
## 97	360021	4
## 98	360051	5
## 99	360052	3
## 100	360111	1
## 101	360112	1
## 102	360113	3
## 103	360114	1
## 104	360121	4
## 105	360131	2
## 106	360132	2
## 107	360141	4
## 108	360161	3
## 109	360181	5
## 110	360191	5
## 111	360211	5
## 112	360221	5
## 113	360222	1
## 114	360223	3
## 115	360251	5
## 116	360301	5
## 117	360302	1
## 118	360331	4
## 119	360341	5
## 120	360351	5
## 121	360352	2
## 122	360353	3
## 123	360361	4
## 124	360362	3
## 125	360371	5
## 126	360372	3
## 127	360391	5
## 128	360392	2
## 129	360393	1
## 130	360411	5
## 131	360471	5
## 132	360472	5
## 133	360481	5
## 134	360482	5
## 135	360491	2
## 136	360501	5
## 137	360502	2
## 138	360511	4
## 139	360531	5
## 140	360541	4
## 141	360551	4
## 142	360561	5
## 143	360562	5
## 144	360571	3

## 145 360581	5
## 146 360591	5
## 147 360592	2
## 148 360611	5
## 149 360621	5
## 150 360671	1
## 151 360672	4
## 152 360673	5
## 153 360674	1
## 154 360691	5
## 155 360701	4
## 156 360711	1
## 157 360721	1
## 158 360741	5
## 159 360751	5
## 160 360752	1
## 161 360781	5
## 162 360782	2
## 163 360791	2
## 164 360801	5
## 165 360811	3
## 166 360812	3
## 167 360813	4
## 168 360841	4
## 169 360842	4
## 170 360871	5
## 171 360881	2
## 172 360901	2
## 173 360911	3
## 174 360912	3
## 175 360921	2
## 176 360941	5
## 177 360951	5
## 178 360952	2
## 179 360991	4
## 180 360992	5
## 181 361001	5
## 182 361002	1
## 183 520021	4
## 184 520022	3
## 185 520031	5
## 186 520041	5
## 187 520042	5
## 188 520051	3
## 189 520061	2
## 190 520062	3
## 191 520063	5
## 192 520071	4
## 193 520072	3
## 194 520081	5
## 195 520091	4

## Compute Average and Quartiles of the Number of Visits for Each Child by Sex:

```
# Calculate the number of visits for each child with non-missing z-scores
visits_per_child_sex <- aggregate(fuvisit ~ id + sex, data=d_clean, FUN=length)

# Calculate the average number of visits by sex
average_visits_by_sex <- aggregate(fuvisit ~ sex, data=visits_per_child_sex, FUN=mean)

# Calculate the quartiles of the number of visits by sex
quartiles_visits_by_sex <- aggregate(fuvisit ~ sex, data=visits_per_child_sex, FUN=function(x) quantile(x, probs=c(0.25, 0.5, 0.75)))

colnames(average_visits_by_sex) <- c("Sex", "Average Visits")
colnames(quartiles_visits_by_sex) <- c("Sex", "Quartiles of Visits")

print(average_visits_by_sex)

##   Sex Average Visits
## 1   1           3.76
## 2   2           3.86

print(quartiles_visits_by_sex)

##   Sex Quartiles of Visits.25% Quartiles of Visits.50% Quartiles of Visits.75%
## 1   1                3                4                5
## 2   2                3                5                5
```

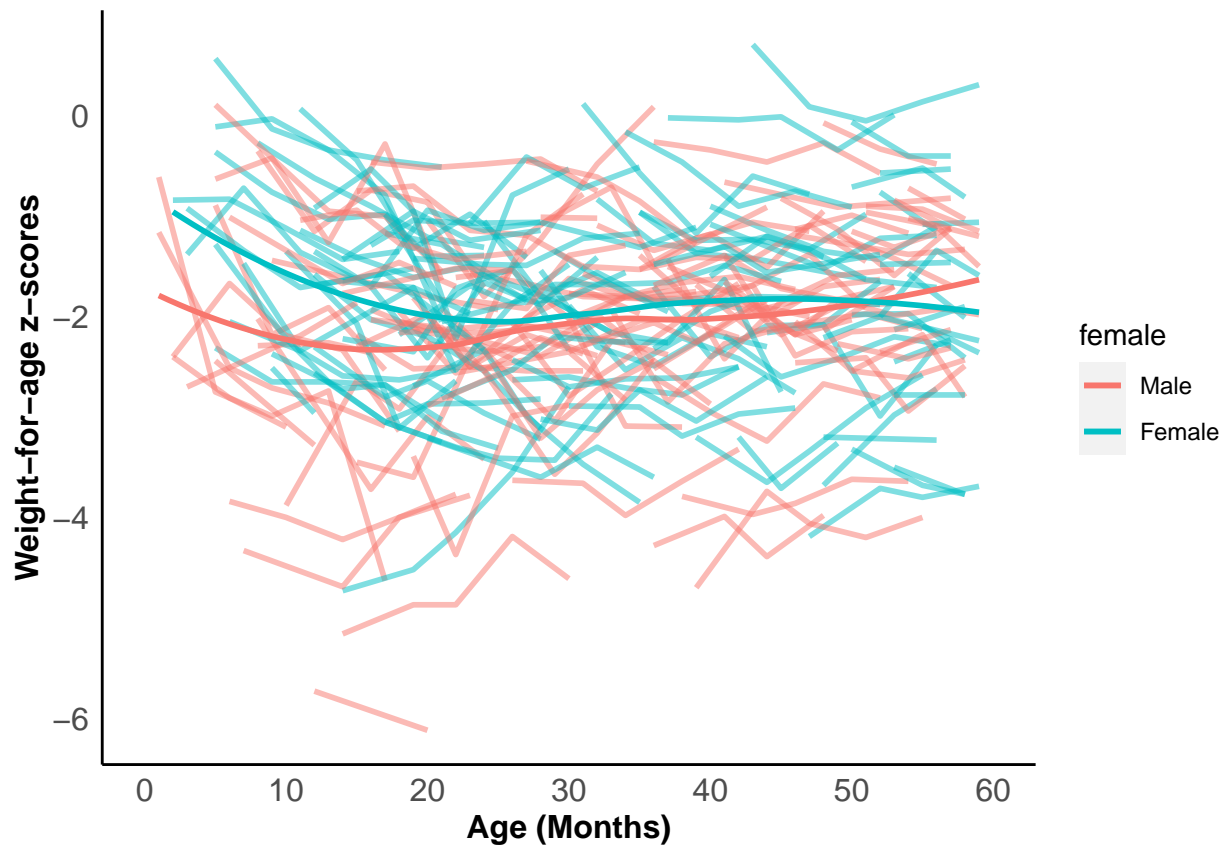
The average of the number of visits is 3.86 for female, and 3.76 for male.

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

```
# Create custom theme
custom_theme <- theme(
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  axis.text = element_text(size = 12),
  axis.title = element_text(size = 12, face = "bold"),
  axis.line = element_line(size = 0.5),
  plot.title = element_text(size = 14, face = "bold", hjust = 0.5)
)

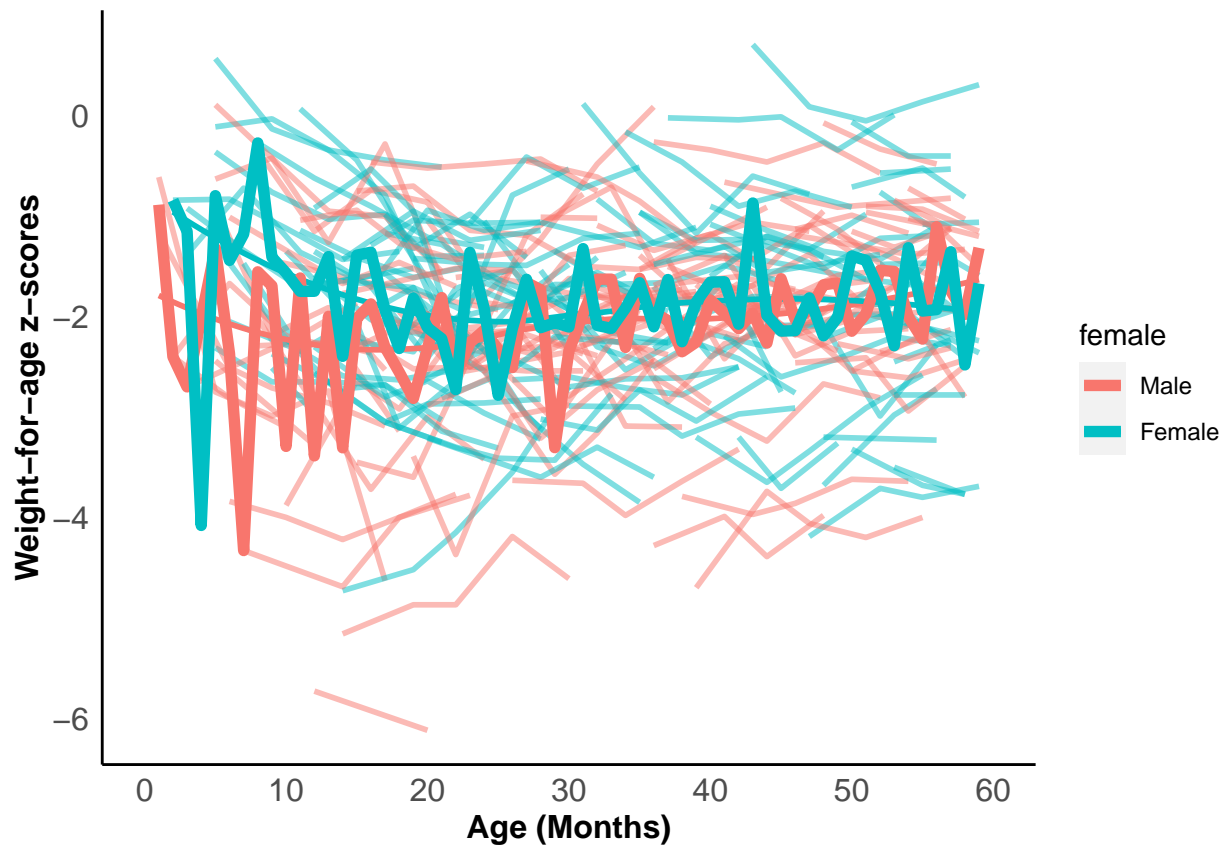
# Spaghetti plot of whole mouth average gingival index
spagplot <- d_clean |>
  ggplot() +
  geom_line(aes(x = age, y = zscores, group = factor(id), color = female), linewidth = 1, alpha = 0.5) +
  geom_smooth(aes(x = age, y = zscores, color = female), method = "loess", se = FALSE) +
  # scale_x_discrete(labels = c("Baseline (1)", "3", "5")) +
  labs(x = "Age (Months)", y = "Weight-for-age z-scores") +
  scale_x_continuous(breaks = seq(0, 60, 10), limits = c(0, 60)) +
  custom_theme
spagplot

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



```
spagplot_mean <- spagplot +
  stat_summary(data = d_clean,
    fun = mean,
    geom = "line",
    aes(x = age, y = zscores, group = female, color = female),
    linewidth = 2)
# scale_color_manual(values = c("#024873", "#920043"),
#   labels = c("Female", "Male"))
spagplot_mean

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



We could see a increase trend in weight-for-age z-scores as age increases in male. We could see a decreasing and then increasing trend of weight-for-age z-scores as age increases in female

## Part II: Model checking and recommendations

```
model11 <- lm(zscores ~ age + female + age*female, data = d_clean )
summary(model11)
```

```
##
## Call:
## lm(formula = zscores ~ age + female + age * female, data = d_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.909 -0.580  0.084  0.656  2.615
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.43506    0.12361  -19.70 < 2e-16 ***
## age             0.01121    0.00338   3.32 0.00094 ***
## femaleFemale    0.69323    0.17316   4.00 6.9e-05 ***
## age:femaleFemale -0.01501    0.00476  -3.15 0.00170 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.993 on 739 degrees of freedom
```



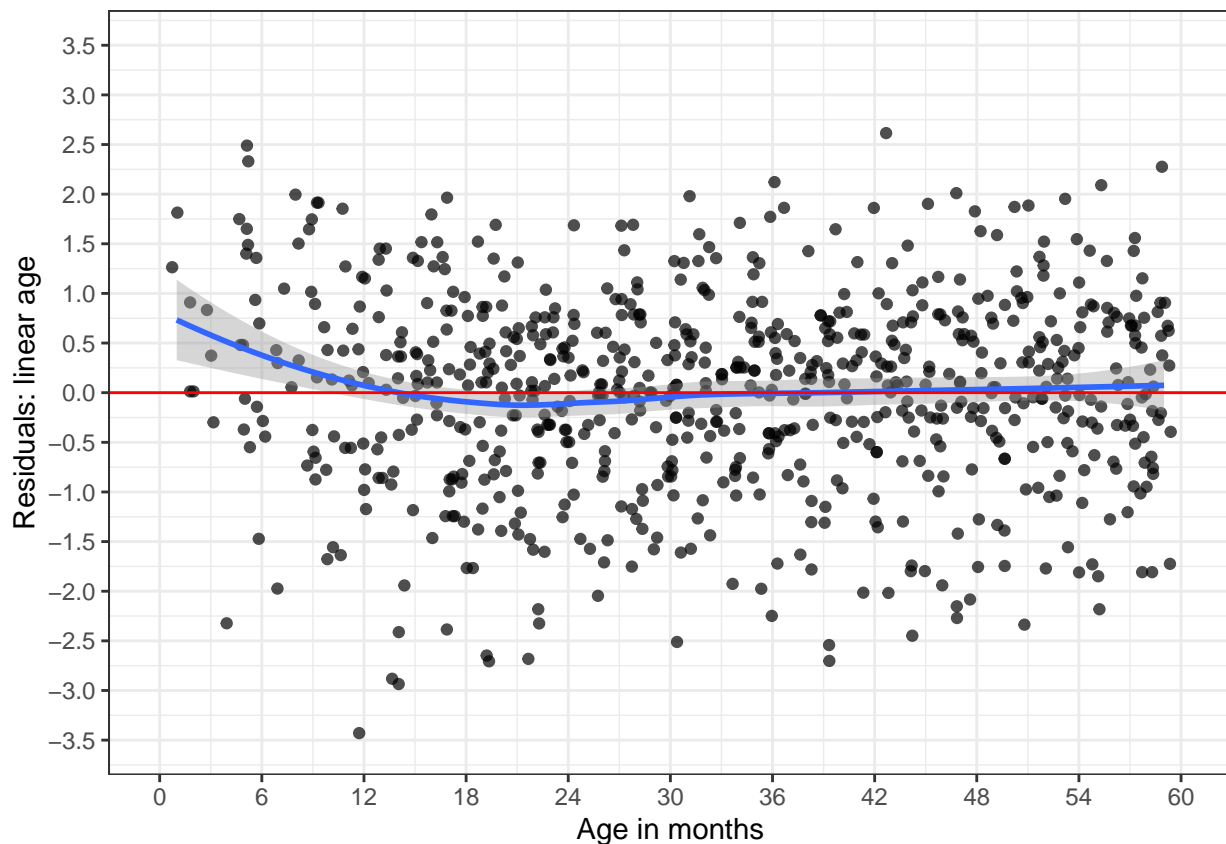
```
## Multiple R-squared:  0.0257, Adjusted R-squared:  0.0217
## F-statistic:  6.5 on 3 and 739 DF,  p-value: 0.000243
```

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.

```
d_clean$residuals = residuals(model1)
ggplot(d_clean, aes(x=age, y=residuals)) +
  geom_jitter(alpha = 0.7) +
  theme_bw() +
  geom_smooth() +
  geom_hline(yintercept=0, color="red") +
  labs(y="Residuals: linear age", x="Age in months") +
  scale_y_continuous(breaks=seq(-3.5, 3.5, 0.5), limits=c(-3.5, 3.5)) +
  scale_x_continuous(breaks=seq(0, 60, 6), limits=c(0, 60))
```

assumption  $E(Y | X) = X \beta$

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



**Independence assumption** Visualizing autocorrelation using a scatterplot matrix

```
# Retrieve only the zscores, age, and Model 1
d_wide <- d_clean |>
  select(residuals, fuvisit, id) |>
  pivot_wider(id_cols = "id",
    names_from = "fuvisit",
```

```

values_from = "residuals")

d_wide <- d_wide |>
  filter(complete.cases(d_wide))

```

```

# View the wide data
head(d_wide, n = 3)

```

```

## # A tibble: 3 x 6
##       id   `0`   `1`   `2`   `3`   `4`
##   <int> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 120021  1.50  1.17  0.903 0.608 0.523
## 2 120051  2.00  1.15  1.52  1.35  0.456
## 3 120052  1.03  0.608 0.554 0.709 0.514

```

```

# Use wide format of the data
cor(d_wide[,c(2:6)])

```

```

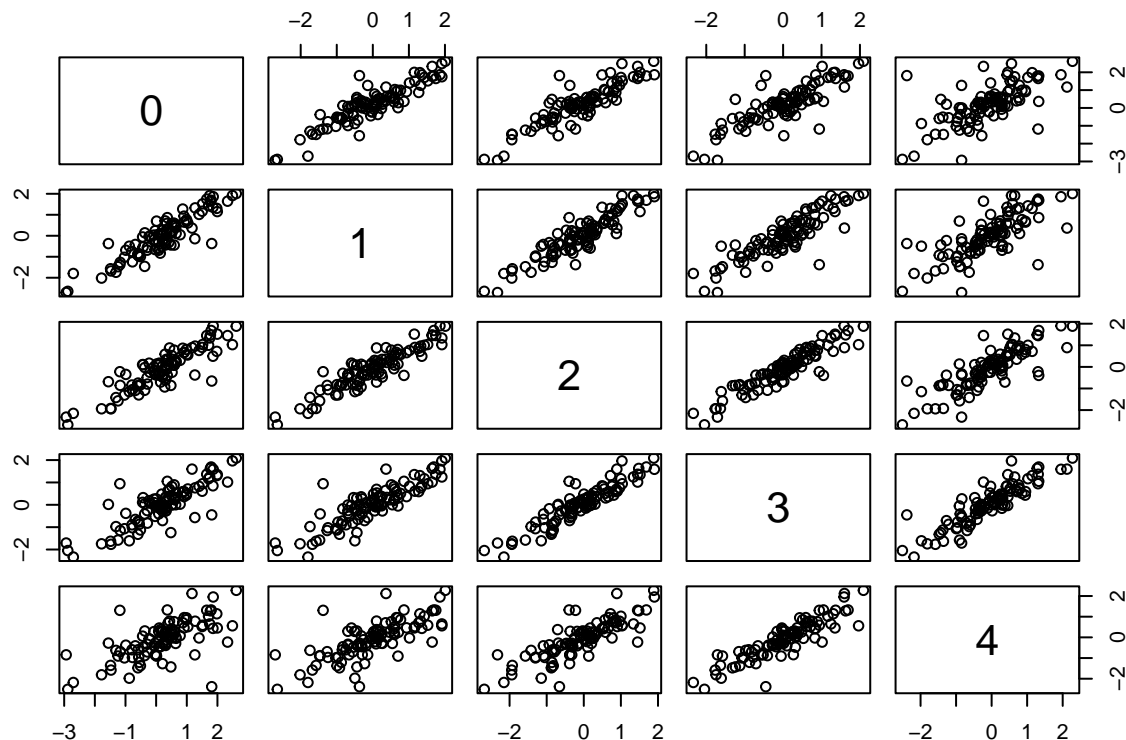
##       0      1      2      3      4
## 0 1.000 0.904 0.883 0.836 0.658
## 1 0.904 1.000 0.920 0.873 0.740
## 2 0.883 0.920 1.000 0.927 0.822
## 3 0.836 0.873 0.927 1.000 0.890
## 4 0.658 0.740 0.822 0.890 1.000

```

```

pairs(d_wide[,c(2:6)])

```



The model for the variance is a function of the visits.

```

# Autocorrelation function
autocorr_fit1 <- gls(zscores ~ age + female + age*female, data = d_clean)
# Run autocorrelation function

```

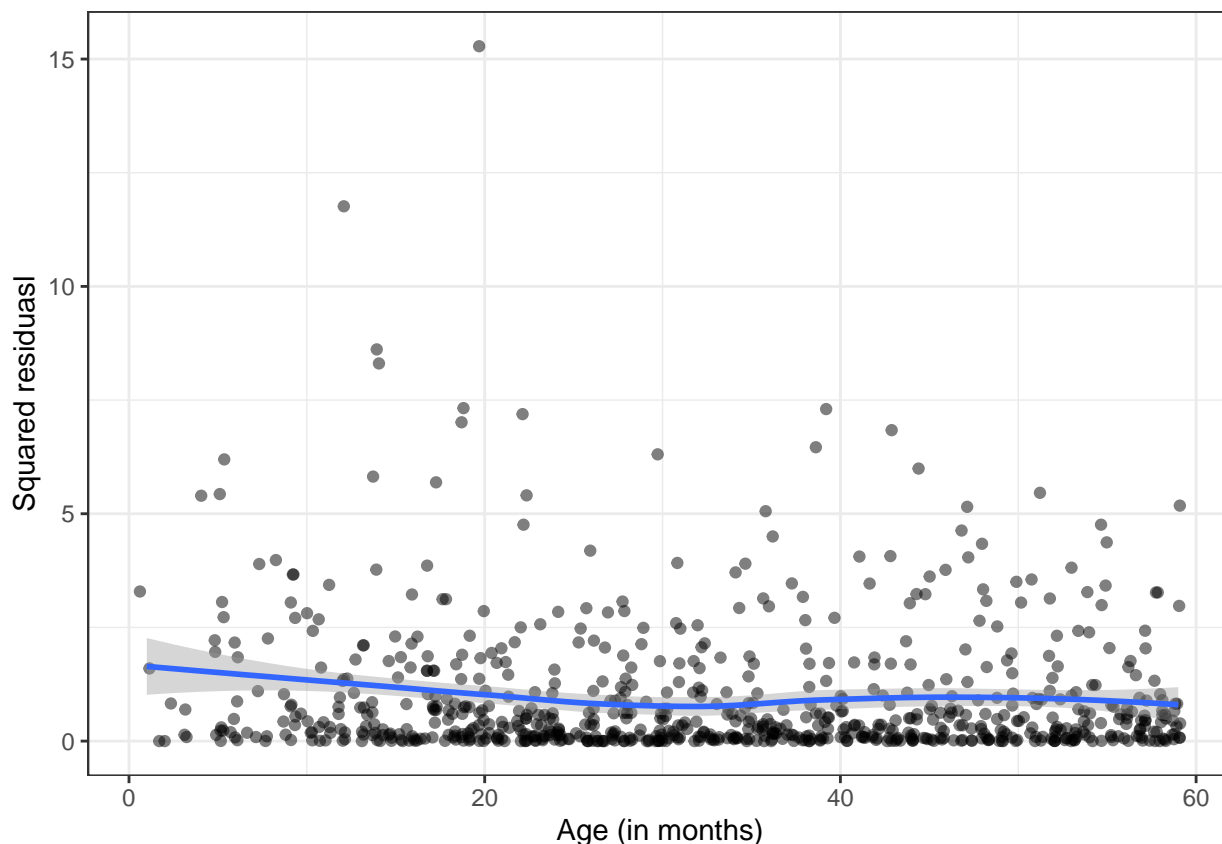
```
# The form argument follows ~ 1 (meaning no covariate) then indicate the ID variable of the individual
ACF(autocorr_fit1, form = ~ 1 | id )
```

```
##   lag   ACF
## 1    0 1.000
## 2    1 0.879
## 3    2 0.819
## 4    3 0.779
## 5    4 0.657
```

```
d_clean = mutate(d_clean, r2 = residuals^2)
# Scatterplot of log squared residuals by age,
# ggplot(d_clean, aes(x=age, y=r2, group = female, color = female)) +
#   geom_jitter(alpha = 0.5) +
#   theme_bw() +
#   geom_smooth() +
#   labs(y="Squared residuasl", x="Age (in months)", color = "Sex")
ggplot(d_clean, aes(x=age, y=r2 )) +
  geom_jitter(alpha = 0.5) +
  theme_bw() +
  geom_smooth() +
  labs(y="Squared residuasl", x="Age (in months)" )
```

constant variance assumptions for the residuals

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



The variation in the residuals for the female children look roughly the same over the age. There appears to be a reduction in the variance of the residuals as age increased to 25 months for male children. So we can propose to modify our multiple linear regression model to relax the homoskedasticity assumption and propose a variance model that is a function of age.

For example our model may be:

$$\text{Var}(\epsilon_{ij}) = \gamma_0 + \gamma_1 |(\text{age} - 25)| (\text{sex} = \text{male})$$

**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 sex 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.**

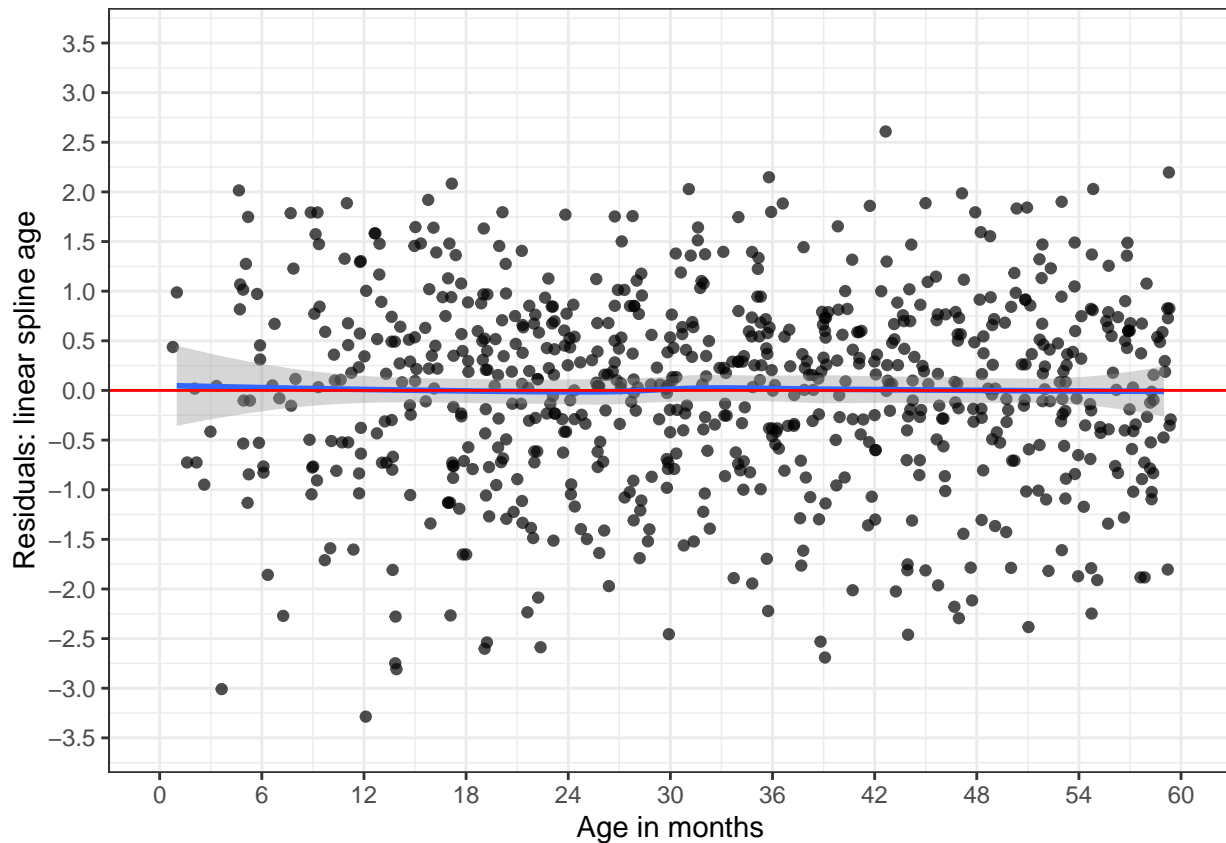
```
d_clean <- d_clean |>
  mutate(age_12 = ifelse(age-12>0, age -12, 0))

model2 <- lm(zscores~ age + age_12 + female + age*female + age_12*female, data = d_clean)
```

assumption  $E(Y | X) = X \beta$  checking the mean:

```
d_clean$residuals2 = residuals(model2)
ggplot(d_clean, aes(x=age, y=residuals2)) +
  geom_jitter(alpha = 0.7) +
  theme_bw() +
  geom_smooth() +
  geom_hline(yintercept=0, color="red") +
  labs(y="Residuals: linear spline age", x="Age in months") +
  scale_y_continuous(breaks=seq(-3.5, 3.5, 0.5), limits=c(-3.5, 3.5)) +
  scale_x_continuous(breaks=seq(0, 60, 6), limits=c(0, 60))
```

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



**Independence assumption** Visualizing autocorrelation using a scatterplot matrix

```
# Retrieve only the zscores, age, and Model 1
```

```
d_wide2 <- d_clean |>
  select(residuals2, fuvisit, id) |>
  pivot_wider(id_cols = "id",
              names_from = "fuvisit",
              values_from = "residuals2")
```

```
# View the wide data
```

```
head(d_wide2, n = 3)
```

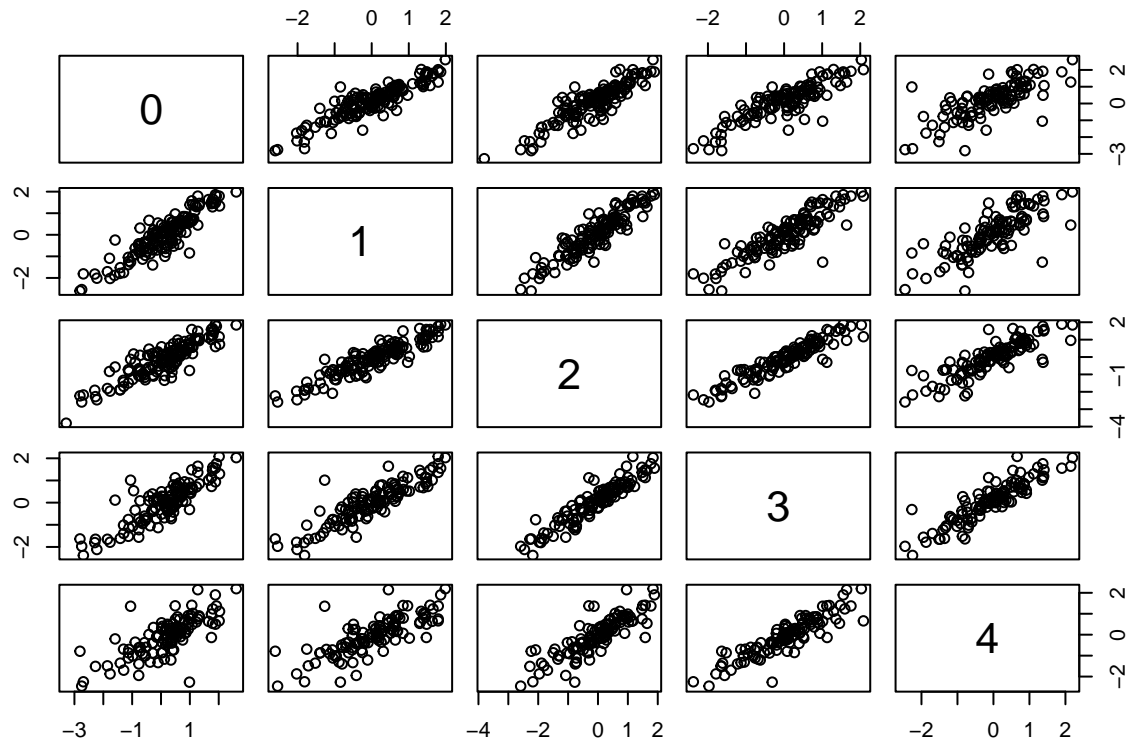
```
## # A tibble: 3 x 6
##       id   `0`   `1`   `2`   `3`   `4`
##   <int> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 120011 0.587 0.193 0.0186 0.0842 NA
## 2 120012 0.678 NA      NA      NA      NA
## 3 120021 1.23  1.30  1.02   0.707  0.604
```

```
# Use wide format of the data
```

```
cor(d_wide2[,c(2:6)])
```

```
##    0  1  2  3  4
## 0  1 NA NA NA NA
## 1 NA  1 NA NA NA
## 2 NA NA  1 NA NA
## 3 NA NA NA  1 NA
## 4 NA NA NA NA  1
```

```
pairs(d_wide2[,c(2:6)])
```



The model for the variance is a function of the visits.

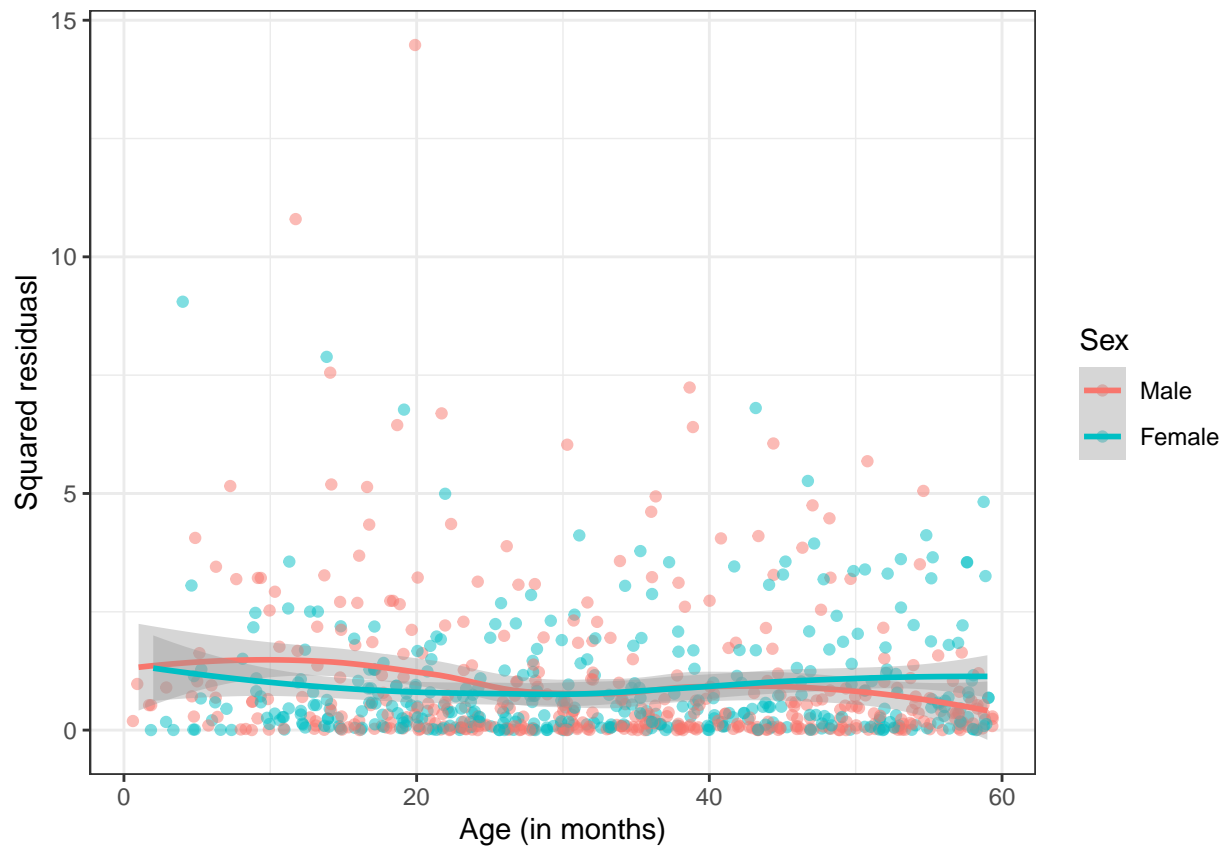
```
# Autocorrelation function
autocorr_fit2 <- gls(zscores ~ age + age_12 + female + age*female + age_12*female, data = d_clean)
# Run autocorrelation function
# The form argument follows ~ 1 (meaning no covariate) then indicate the ID variable of the individual
ACF(autocorr_fit2, form = ~ 1 | id )
```

```
## lag ACF
## 1 0 1.000
## 2 1 0.883
## 3 2 0.829
## 4 3 0.803
## 5 4 0.713
```

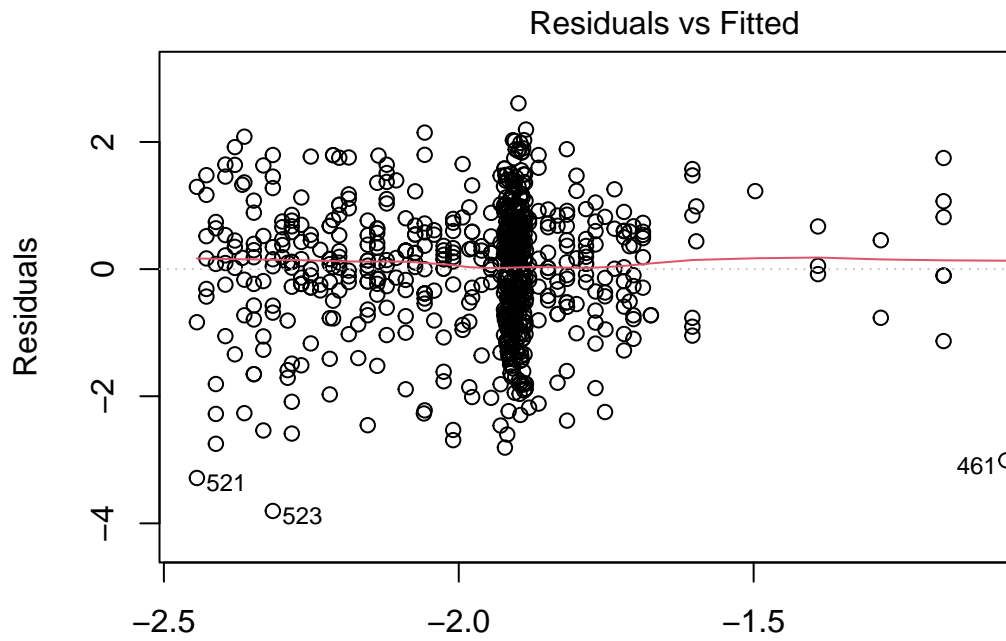
```
d_clean = mutate(d_clean, r2_new = residuals^2)
# Scatterplot of log squared residuals by age,
ggplot(d_clean, aes(x=age, y=r2_new, group = female, color = female)) +
  geom_jitter(alpha = 0.5) +
  theme_bw() +
  geom_smooth() +
  labs(y="Squared residuas1", x="Age (in months)", color = "Sex")
```

constant variance assumptions for the residuals

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



```
plot(model2)
```

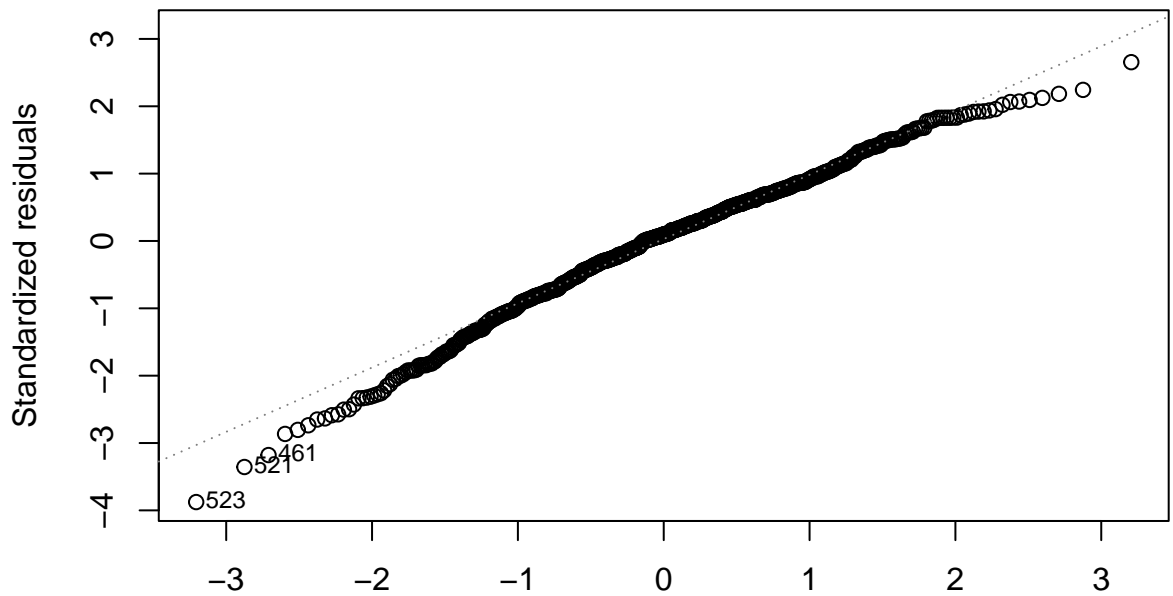


residual are normally distributed

Fitted values

$\text{lm}(\text{zscores} \sim \text{age} + \text{age\_12} + \text{female} + \text{age} * \text{female} + \text{age\_12} * \text{female})$

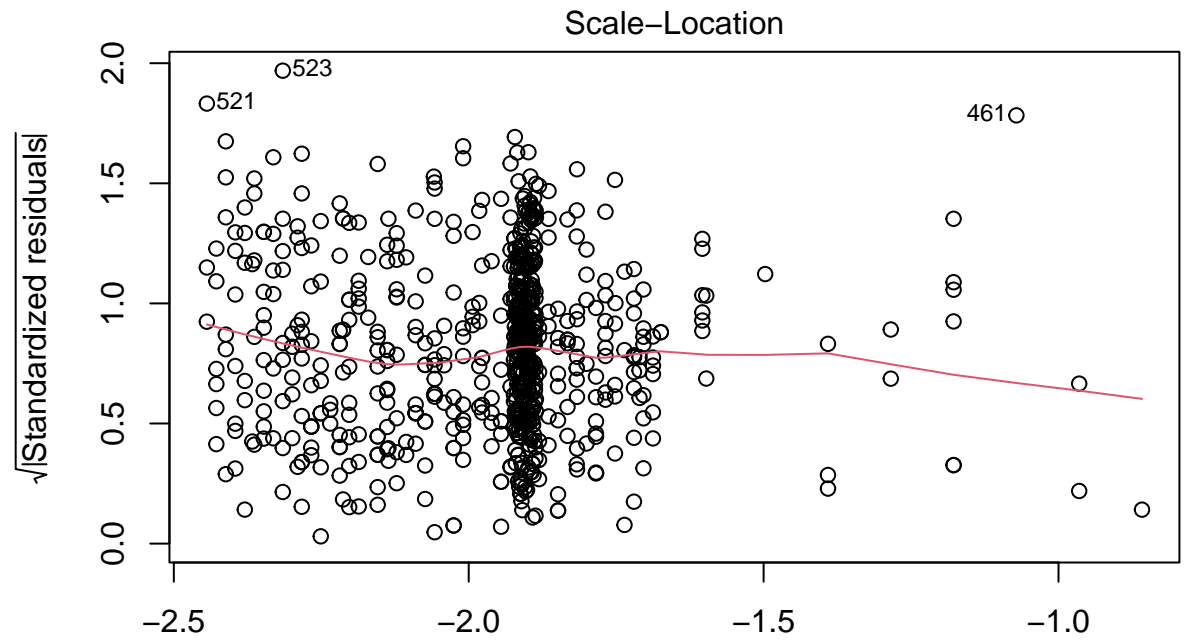
Q-Q Residuals



Theoretical Quantiles

$\text{lm}(\text{zscores} \sim \text{age} + \text{age\_12} + \text{female} + \text{age} * \text{female} + \text{age\_12} * \text{female})$





### Part III: Marginal model for longitudinal data

1. Use the `gls` function in R to fit the model you proposed in Part I.

```
# AR(1):
model_AR <- gls(zscores ~age + age_12 + female + age*female + age_12*female, data = d_clean, correlation = 0.9)
summary(model_AR)$tTable

##               Value Std.Error t-value  p-value
## (Intercept)   -1.17588    0.2282  -5.154 3.28e-07
## age           -0.09256    0.0180  -5.134 3.64e-07
## age_12         0.10202    0.0191   5.340 1.24e-07
## femaleFemale   0.35031    0.4144   0.845 3.98e-01
## age:femaleFemale 0.00896    0.0333   0.269 7.88e-01
## age_12:femaleFemale -0.02286    0.0351  -0.651 5.15e-01

# Toeplitz:
# model_Toep <- gls(zscores ~age + age_12 + female + age*female + age_12*female, data = d_clean, correlation = 0.9)
# summary(model_Toep)$tTable
```

2. From the fit of the model, compute the estimated  $\text{Corr}(\epsilon_{i1}, \epsilon_{ij})$  for  $j = 2, 3, 4, 5$  where the follow-up visits (fuvisit) have values 0 (baseline,  $j=1$ ) and 1, 2, 3, 4 (representing the 4 follow-up visits each 4 months apart,  $j = 2, 3, 4, 5$ ). Compare these model-based correlation estimates to those you computed in Part II Question 1.

```
# AR(1):
female.V = getVarCov(model_AR, individual=3)
cov2cor(female.V)

## Marginal variance covariance matrix
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 1.000 0.912 0.831 0.758 0.691
## [2,] 0.912 1.000 0.912 0.831 0.758
## [3,] 0.831 0.912 1.000 0.912 0.831
## [4,] 0.758 0.831 0.912 1.000 0.912
## [5,] 0.691 0.758 0.831 0.912 1.000
## Standard Deviations: 1 1 1 1 1

male.V = getVarCov(model_AR, individual=7)
cov2cor(male.V)

## Marginal variance covariance matrix
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 1.000 0.912 0.831 0.758 0.691
## [2,] 0.912 1.000 0.912 0.831 0.758
## [3,] 0.831 0.912 1.000 0.912 0.831
## [4,] 0.758 0.831 0.912 1.000 0.912
## [5,] 0.691 0.758 0.831 0.912 1.000
## Standard Deviations: 1 1 1 1 1

# Toeplitz:
# getVarCov(model_Toep, individual=1)
# round(cov2cor(getVarCov(model_Toep, individual=1)), 3)
```

For time point 1 (baseline,  $j=1$ ) to time point 2 ( $j=2$ ), the AR(1) model estimates correlation of 0.912, while the previously computed correlation was 0.904. For time point 1 (baseline,  $j=1$ ) to time point 3 ( $j=3$ ), the AR(1) model estimates the correlation as 0.831, in comparison to the previously one 0.883. Correlation between time point 1 and 4 was 0.758 in AR(1) model, and 0.836 in previous model. This AR(1) model provides a more smooth decrease in correlation over time, which is characteristic of the exponential decay pattern typical of AR(1) processes.

3. Conduct a Wald test to address the overall goal of the analysis; i.e. to determine if the average growth rates of children differ by sex.

```
library(nlme)

# Extract estimated regression coefficients
beta_hat <- coef(model_AR)

# estimated variance of  $\beta$ 
V_beta_hat <- summary(model_AR)$varBeta

C <- matrix(c(0, 0, 0, 1, 0, 0, # for female
              0, 0, 0, 0, 1, 0, # for age:female
              0, 0, 0, 0, 0, 1), # for age_12:female
            nrow = 3, byrow = TRUE)

# Calculate the Wald test statistic Q
Q <- t(C %*% beta_hat) %*% solve(C %*% V_beta_hat %*% t(C)) %*% (C %*% beta_hat)

# Degrees of freedom for the chi-squared distribution
df <- nrow(C)

# Calculate p-value from the chi-squared distribution
p_value <- 1 - pchisq(Q, df)

# Output the test statistic and p-value
list(Wald_test_statistic = Q, p_value = p_value)

## $Wald_test_statistic
##      [,1]
## [1,] 4.74
##
## $p_value
##      [,1]
## [1,] 0.192
```

The Wald test statistic is 4.74 and p-value is 0.192. We fail to reject the null hypothesis that the average growth rates are the same by sex.

## Part IV: Sensitivity analysis for the marginal model results

1.

a. Show that the robust variance estimator is equal to the generalized least squares variance estimator when  $\epsilon$  is correctly specified

```
# install.packages("clubSandwich")
library(clubSandwich)
```

b. For the model you fit in Part III Question1, obtain the robust standard error estimates using the Huber-White sandwich estimator. Compare the estimated model based and robust standard errors.

a.  $\Sigma_w$  is correctly specified, ie  $\varepsilon \sim MVN(0, \Sigma_w = \Sigma)$

So  $\hat{\Sigma}_w = \Sigma$

$$\hat{\Sigma}_E = (Y - X\hat{\beta}_{gls})(Y - X\hat{\beta}_{gls})' = \varepsilon\varepsilon'$$

since  $\varepsilon \sim MVN(0, \Sigma_w = \Sigma)$ ,  $\varepsilon\varepsilon' = \Sigma$

plug in  $\hat{\Sigma}_w = \Sigma$  and  $\hat{\Sigma}_E = \Sigma$  into  $\text{Var}(\hat{\beta}_{gls})_{\text{robust}}$

$$\text{Var}(\hat{\beta}_{gls})_{\text{robust}} = (X' \hat{\Sigma}_w^{-1} X)^{-1} X' \hat{\Sigma}_w^{-1} \hat{\Sigma}_E \hat{\Sigma}_w^{-1} X (X' \hat{\Sigma}_w^{-1} X)^{-1}$$

$$= (X' \hat{\Sigma}_w^{-1} X)^{-1} X' \Sigma^{-1} \Sigma \Sigma^{-1} X (X' \hat{\Sigma}_w^{-1} X)^{-1}$$

$$= (X' \hat{\Sigma}_w^{-1} X)^{-1} X' \Sigma^{-1} X (X' \hat{\Sigma}_w^{-1} X)^{-1}$$

$$= (X' \hat{\Sigma}_w^{-1} X)^{-1} \Rightarrow \text{gls variance estimator}$$

So the robust variance estimator is equal to the gls variance estimator

Figure 1: Part 4-1

```
## Registered S3 method overwritten by 'clubSandwich':
```

```
## method from
```

```
## bread.mlm sandwich
```

```
# Fit the model
```

```
model_AR <- gls(zscores ~age + age_12 + female + age*female + age_12*female, data = d_clean, correlation =
```

```
# Estimate robust standard errors using the cluster-robust sandwich estimator
```

```
# This is the robust estimate of Var-hat(beta-hat)
```

```
vcov.rob <- vcovCR(model_AR, cluster = d_clean$id, type = "CRO")
```

```
# Save the results for testing each individual coefficient
```

```
clubsand <- coef_test(model_AR, vcov = vcov.rob)
```

```
# Compare the standard errors
```

```
summary(model_AR)$tTable
```

```
##              Value Std.Error t-value p-value
## (Intercept)  -1.17588    0.2282  -5.154 3.28e-07
## age          -0.09256    0.0180  -5.134 3.64e-07
## age_12        0.10202    0.0191   5.340 1.24e-07
## femaleFemale  0.35031    0.4144   0.845 3.98e-01
## age:femaleFemale 0.00896    0.0333   0.269 7.88e-01
## age_12:femaleFemale -0.02286    0.0351  -0.651 5.15e-01
```

```
clubsand
```

```
##              Coef. Estimate    SE t-stat d.f. (Satt) p-val (Satt) Sig.
## (Intercept) -1.17588 0.3509 -3.351    16.0    0.00407 **
## age         -0.09256 0.0276 -3.356    13.0    0.00518 **
## age_12       0.10202 0.0281  3.636    15.4    0.00235 **
## femaleFemale 0.35031 0.4340  0.807    26.3    0.42681
## age:femaleFemale 0.00896 0.0331  0.271    21.5    0.78895
```

```
## age_12:femaleFemale -0.02286 0.0337 -0.679      25.1      0.50343
```

For intercept, The model-based standard error is 0.2282, while the robust standard error is 0.3509. For “age” term, the model-based standard error is 0.0180, and the robust is 0.0276. For “age\_12” and “female” terms, the model-based standard error are all smaller than the robust one. The robust standard errors are larger than the model-based standard errors, this suggests that the model-based errors may be underestimating the true variability in the coefficients due to potential violations of the model assumptions (such as non-constant variance or correlations not being correctly modeled).

**c. Using the results of a. and b. above, do the data support or not support your working model for the variance/covariance of the residuals?** The robust standard errors for most coefficients are larger than the model-based standard errors. That suggests that the working model assumptions for the variance/covariance of the residuals may not be fully appropriate. For the interaction terms, the robust and model-based standard errors are very similar, suggesting that the working model may adequately capture the variance/covariance structure for these specific terms. In conclusion, the data do not fully support the working model for the variance/covariance of the residuals.

```
# Calculate the Wald test statistic using the robust variance-covariance matrix
Q_robust <- t(C %*% coef(model_AR)) %*% solve(C %*% vcov.rob %*% t(C)) %*% (C %*% coef(model_AR))

# Degrees of freedom: number of restrictions being tested, equal to the number of rows in C
df_robust <- nrow(C)

# Calculate the p-value from the chi-squared distribution
p_value_robust <- 1 - pchisq(Q_robust, df_robust)

# Output the Wald test statistic and p-value
list(Wald_test_statistic_robust = Q_robust, p_value_robust = p_value_robust)
```

**d. Use the robust variance estimate for you obtained (called vcov.rob in the code above) and repeat the Wald test you conducted in Part III Question 3. Are the results of the Wald tests the same or different?**

```
## $Wald_test_statistic_robust
##      [,1]
## [1,] 6.01
##
## $p_value_robust
##      [,1]
## [1,] 0.111
```

Use the robust variance estimate, the Wald test statistic is 6.01 and p-value is 0.111. We fail to reject the null hypothesis that the average growth rates are the same by sex. The statistic are different from the Wald test conducted using the variance estimated by the GLS model, but they both fail to reject the null.

**2. Instead of modelling the variance/covariance of the within subject residuals, you could assume an independence working model with constant variance, i.e. fit the model using ordinary least squares, and apply a robust variance estimate for inference on .**

```
# Fit the ordinary least squares model
fit.ols <- lm(zscores~ age + age_12 + female + age*female + age_12*female, data = d_clean)
# Get the robust variance estimate
```

```
vcov.rob.ols <- vcovCR(fit.ols, cluster = d_clean$id, type = "CRO")
# Save the results for testing each individual coefficient
clubsand.ols <- coef_test(fit.ols, vcov = vcov.rob.ols)
# Compare the standard errors for the estimated coefficients
summary(fit.ols)$coeff
```

a. Use the `lm` command to refit your model under working independence and constant variance; then obtain a robust variance estimate for

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.5203    0.3660  -4.154 3.65e-05
## age           -0.0770    0.0334  -2.304 2.15e-02
## age_12         0.0931    0.0351   2.653 8.16e-03
## femaleFemale   0.8748    0.5533   1.581 1.14e-01
## age:femaleFemale -0.0295    0.0501  -0.589 5.56e-01
## age_12:femaleFemale 0.0142    0.0524   0.272 7.86e-01
```

```
clubsand.ols
```

```
##              Coef. Estimate      SE t-stat d.f. (Satt) p-val (Satt) Sig.
##      (Intercept)  -1.5203 0.3639  -4.178      9.57    0.00209  **
##              age   -0.0770 0.0338  -2.279     12.13    0.04155   *
##      age_12        0.0931 0.0370   2.513     13.86    0.02499   *
##      femaleFemale  0.8748 0.6017   1.454     20.23    0.16129
##      age:femaleFemale -0.0295 0.0545  -0.542     25.21    0.59261
##      age_12:femaleFemale 0.0142 0.0584   0.243     28.46    0.80948
```

In the model using ordinary least squares, the robust standard error for intercept is 0.3639. For “age” term, the robust standard error is 0.0338. The ols robust standard errors are similar to the model-based standard errors.

```
# Calculate the Wald test statistic using the robust variance-covariance matrix
Q_robust_ols <- t(C %%% coef(fit.ols)) %%% solve(C %%% vcov.rob %%% t(C)) %%% (C %%% coef(fit.ols))

# Degrees of freedom: number of restrictions being tested, equal to the number of rows in C
df_robust <- nrow(C)

# Calculate the p-value from the chi-squared distribution
p_value_robust_ols <- 1 - pchisq(Q_robust_ols, df_robust)

# Output the Wald test statistic and p-value
list(Wald_test_statistic_robust_ols = Q_robust_ols, p_value_robust_ols = p_value_robust_ols)
```

b. Recalculate the Wald test using the robust variance estimate for from the working independence and constant variance model.

```
## $Wald_test_statistic_robust_ols
##      [,1]
## [1,] 8.43
##
## $p_value_robust_ols
##      [,1]
## [1,] 0.0379
```

Use the robust variance estimate, the Wald test statistic is 8.43 and p-value is 0.0379.

c. Compare the estimated standard errors for and the Wald test results based on your three approaches: i) assuming your working model is correct (Part III), ii) a robust variance estimate applied to your working model (Part IV Question 1), and iii) a robust variance estimate applied to a working independence/constant variance model (Part IV Question 2).

- i) assuming your working model is correct, the Wald test statistic is 4.74 and p-value is 0.192, This result suggests that there is not enough statistical evidence to reject the null hypothesis of no sex difference in average growth rates.
- ii) Using a robust variance estimate applied to the working model (GLS robust variance estimate), the Wald test statistic increased to 6.01, and the p-value decreased to 0.111. But still did not show a significant sex difference in average growth rates.
- iii) Using a robust variance estimate applied to a working independence/constant variance model (OLS robust variance estimate), the Wald test statistic is 8.43, and the p-value is 0.0379, suggesting reject the null hypothesis, indicating that there is a sex difference in average growth rates.

The results indicate increasing evidence against the null hypothesis when robust variance estimates are used, with the OLS robust estimate leading to a rejection of the null hypothesis. This progression suggests potential model misspecification and the impact of heteroscedasticity on inference.

**3. The bootstrap procedure can also be applied to longitudinal or clustered data to estimate standard errors of estimated coefficients (or functions of).**

## 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 <- d[,c('id','age','zscores','female','fuvisit')] %>% pivot_wider(id_cols=c(id,female),values

## 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,female),
                        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(zscores ~age + I(I(age>=12)*(age-12)) + female +
          age:female +
          I(I(age>=12)*(age-12)):female, dlong)
```

```

  coefficients(fit)
}

result = boot(nepal.wide, my.boot, 1000)
boot.V <- cov(result$t)
boot.se <- sqrt(diag(boot.V))
boot.se

```

a. Compute the bootstrap standard error estimates and compare these to the standard errors you obtained in the three earlier approaches, i.e. i, ii, and iii defined above. Comment on similarities and differences.

```
## [1] 0.4041 0.0374 0.0406 0.6571 0.0586 0.0622
```

Using bootstrap procedure, the bootstrap standard error estimates are 0.4000, 0.0366, 0.0396, 0.6574, 0.0581, 0.0615 for the intercept, “age”, “age\_6”, female”, “agefemale”, “age\_6female”. Compared to the three earlier approaches, bootstrap standard error is larger than gls robust standard error estimates using the Huber-White sandwich estimator and ols robust standard errors.

```

beta_hat = result$t0
var_beta_hat = boot.V
C <- matrix(c(0,0,0,1,0,0,
              0,0,0,0,1,0,
              0,0,0,0,0,1),
            ncol = 6, nrow = 3, byrow = TRUE)
Q_bootstrap <- t(C %*% beta_hat) %*% solve(C %*% var_beta_hat %*% t(C)) %*% (C %*% beta_hat)

df_bootstrap <- nrow(C)

# Calculate the p-value from the chi-squared distribution
p_value_bootstrap <- 1 - pchisq(Q_bootstrap, df_bootstrap)

# Output the Wald test statistic and p-value
list(Wald_test_statistic_bootstrap = Q_bootstrap, p_value_bootstrap = p_value_bootstrap)

```

b. Repeat the Wald test using the bootstrap estimate of the variance of . NOTE: you can use Comment on similarities and differences.

```

## $Wald_test_statistic_bootstrap
##      [,1]
## [1,] 5.64
##
## $p_value_bootstrap
##      [,1]
## [1,] 0.131

```

the Wald test statistic is 5.16 and p-value is 0.161

## Part V: Summarize your findings

Write a brief report (no more than 1000 words) with sections: objective, data, methods, results, summary as if for a health services journal.