Comparing Models for Single-Cell RNA Sequencing Data

Final Project -- BIOS 6643 – FALL 2019

Lee Panter

Introduction

The discovery of DNA, RNA, and the development of genomic sequencing methods has proven to be a valuable tool of scientific research. However, the methods developed in the first-generation of technology relied on “bulk” sampling, which could only estimate population-average expression of RNA and DNA. In order to obtain a more complete understanding of how the cellular landscape functions, estimates of cell-to-cell variability would need to be estimated. Development of single-cell RNA sequencing (scRNAseq) technology has increased to satisfy this need, but a need for statistical analysis is still outstanding since the previous methods used to model bulk RNA sequencing data do not account for the correlated nature of scRNAseq data.

This paper will compare seven different modeling approaches on an observational scRNAseq data set obtained from a Lupus Nephritis Case/Control Study involving 33 patients across the United States. (Arazi et al. 2018) Two RNA genes were selected to be the predictor-response pair to simplify the modeling process. The main goal of this report is to investigate the ways in which parameter estimates vary as the modeling methodology is altered. It is hoped that the results of this investigation are useful for the development of future models for scRNAseq data sets.

Data

A single-cell RNA sequencing (scRNAseq) expression profile is a matrix of count-values representing a “snapshot” of the magnitude of activity of genomic features of a single cell. (“Gene Expression Profiling - Wikipedia,” n.d.) In its original form, the data being studied here had a data matrix that contained 9,560 single-cell observations clustered within 27 samples (5 control not included in data). Each observation contained the expression of 38,354 genetic features.

Single-Cell data is often unreliable, protocol dependent, and can often have batch effects. Data quality control (QC) filters out redundant measures, and dead cell observations. The Seurat guided tutorial (Satija and others 2018) was used to perform quality control, using the parameter values:

- Percent Mitochondrial DNA > 60%
- Genetic Features Expressed < 1,000
- Genetic Features Expressed > 5,000
- B-cells only

These quality control measures reduced the original data by 88%, leaving only 1,110 observations clustered within 15 samples. Two genes (MALAT1 and CD19) were then selected from the set of genetic features in the initial data to be studied due to a higher correlation. MALAT1 has been linked with cancer metastasis, cell migration, and cell regulation. (“MALAT1 Gene - Genecards | Malat1 Rna Gene,” n.d.) CD19 encodes a cell surface molecule which regulates lymphocyte proliferation and differentiation. (“CD19 Gene - Genecards | Cd19 Protein | Cd19 Antibody,” n.d.).

Histograms and a joint distribution scatter plot were constructed to visualize the distributions of the selected variables (Appendix: Fig1-Fig3). The presence of zeros in the data indicated that the distribution might be well suited for a zero-inflated mixture model. Specifically, since the response was count-valued, the histograms indicated that a zero-inflated Poisson Generalized Linear Model or Generalized Linear Mixed Model would be appropriate. Additionally, while normality was not expected, log-transformations were also applied (Appendix: fig4-fig6), and resulted in approximate normality of the response MALAT1 and a bimodal distribution of the predictor CD19.

Methods

The seven different modeling methodologies explored in this investigation are:

1. Linear Models with Fixed Effects (LMwFE)
2. Linear Mixed Models with Random Effects (LMMwRE)
3. Poisson Generalized Linear Models without overdispersion (POI)
4. Poisson quasi-likelihood Generalized Linear Models with over dispersion (POIql)
5. Poisson Generalized Linear Mixed Models fit using Penalized Quasi-Likelihood (POIlmm)
6. Zero-Inflated Poisson Generalized Linear Mixed Model – Fixed Effect Subject (ZIPwFE)
7. Zero-Inflated Poisson Generalized Linear Mixed Model – Random Effect Subject (ZIPwRE)

Models (1) and (2) are fit on log-transformed data (predictor and response), the remaining models are fit on untransformed data.

We assume that repeated measure residual errors are independent (when applicable), i.e for subject $i = 1, \ldots, 15$ and repeated measure $j = 1 \ldots, n_i$

$$\epsilon_{ij} \sim N(0, \sigma_\epsilon^2)$$

In an attempt to gain insight into the processes governing cellular relationships with their host subjects, a subject parameter was also fit using either a fixed or a random effect.

$$\begin{aligned} Fixed\ Effect: &\quad \beta_{0i} \\ Random\ Effect: &\quad b_{0i} \end{aligned}$$

where:

$$b_{0i} \sim N(0, \sigma_b^2)$$

A fixed effect, global intercept and a covariate (CD19) parameter are included in all models, these parameter estimates (along with their standard errors) will be the primary focus for comparing the modeling methods.

We let:

$$\gamma_{ij} = \beta_0 + \beta_1\ CD19$$

and unless otherwise stated, in situations where a model calls for a link function, the canonical link function for count data will be used:

$$\mu_{ij} = g^{-1}(\eta_{ij}) = log(\eta_{ij})$$

Zero-Inflated models will use a fixed effect intensity model with only an intercept

$$R_{ij} \sim bernoulli(p_{ij}|\alpha_{0i}) \quad p(R_{ij} = 1) = \alpha_{0i}$$

the intensity model parameters will be varied using a fixed or random effect for subject. This terminology allows the models to be written as:

Model #	Model Name	Model Format	Assumed Data Distribution
1	LMwFE	$Y_{ij} = \beta_{0i} + \gamma_{ij} + \epsilon_{ij}$	$Y_{ij} \sim N\left(\beta_{0i} + E\left[\ \gamma_{ij}\right], \sigma_\epsilon^2\right)$
2	LMMwRE	$Y_{ij} = \gamma_{ij} + b_{0i} + +\epsilon_{ij}$	$E\left[Y_{ij} \mid b_{0i}\right] \sim N\left(E\left[\ \gamma_{ij} + b_{0i}+\right], \sigma_\epsilon^2\right)$
3	POI	$\mu_{ij} = g^{-1}\left(\eta_{ij} = \beta_{0i} + \gamma_{ij}\right)$	$Y_{ij} \sim Poisson\left(\beta_{0i} + E\left[\ \gamma_{ij}\right]\right)$
4	POIql	$\mu_{ij} = g^{-1}\left(\eta_{ij} = \beta_{0i} + \gamma_{ij}\right)$	$Y_{ij} \sim Poisson\left(\phi\left(\beta_{0i} + E\left[\ \gamma_{ij}\right]\right)\right)$
5	POIlmm	$\mu_{ij} = g^{-1}\left(\eta_{ij} = \gamma_{ij} + b_{0i}\right)$	$E\left[Y_{ij} \mid b_{0i}\right] \sim Poisson\left(E\left[\ \gamma_{ij}\right] + b_{0i}\right)$
6	ZIPwFE	$R_{ij} \sim bernoulli(p_{ij} \mid \alpha_{ij})$ $\mu_{ij} \mid (r_{ij} = 1) = g\left(\eta_{ij} = \beta_{0i} + \gamma_{ij}\right)$	$Y_{ij} \sim ZerInfPoi\left(\beta_{0i} + E\left[\ \gamma_{ij}\right], p_{ij}\right)$
7	ZIPwRE	$R_{ij} \sim bernoulli(p_{ij} \mid \alpha_{ij})$ $\mu_{ij} \mid (r_{ij} = 1) = g\left(\eta_{ij} = \gamma_{ij} + b_{0i}\right)$	$E\left[Y_{ij} \mid b_{0i}\right] \sim ZerInfPoi\left(E\left[\ \gamma_{ij}\right] + b_{0i}, p_{ij}\right)$

Results

Intercept Estimates

Model	Estimate	Std. Error	pvalue
LMwFE	8.3464	$4.981 * 10^{-2}$	$< 2 * 10^{-16}$
LMMwRE	8.3479	$1.3565 * 10^{-1}$	$< 2 * 10^{-16}$
POI	8.821	$4.856 * 10^{-4}$	$< 2 * 10^{-16}$
POIql	8.957	$3.007 * 10^{-2}$	$< 2 * 10^{-16}$
POIlmm	8.8362	$1.0163 * 10^{-1}$	$< 1 * 10^{-5}$
ZIPfe	8.958	$3.689 * 10^{-4}$	$< 2 * 10^{-16}$
ZIPre	8.9402	$6.5229 * 10^{-4}$	$< 1 * 10^{-4}$

Slope Estimates

Model	Estimate	Std. Error	pvalue
LMwFE	$5.590 * 10^{-2}$	$1.534 * 10^{-2}$	$2.82 * 10^{-4}$
LMMwRE	$5.703 * 10^{-2}$	$1.528 * 10^{-1}$	$1.8935 * 10^{-4}$
POI	$3.246 * 10^{-4}$	$2.513 * 10^{-6}$	$< 2 * 10^{-16}$
POIql	$8.839 * 10^{-5}$	$1.913 * 10^{-4}$	0.644
POIlmm	$3.16 * 10^{-4}$	$1.6525 * 10^{-4}$	$5.61 * 10^{-2}$
ZIPfe	$8.559 * 10^{-5}$	$2.176 * 10^{-6}$	$< 2 * 10^{-16}$
ZIPre	$2.9282 * 10^{-4}$	$2.1289 * 10^{-6}$	$1 * 10^{-4}$

Intercept Estimate Percent Change

	1	2	3	4	5	6	7
1	0	0.00	-0.06	-0.06	-0.06	-0.06	-0.07
2	0.00	0	-0.06	-0.06	-0.06	-0.06	-0.07
3	0.05	0.05	0	0.00	-0.00	0.00	-0.01
4	0.05	0.05	0.00	0	-0.00	0.00	-0.01
5	0.06	0.06	0.00	0.00	0	0.00	-0.01
6	0.05	0.05	0.00	0.00	-0.00	0	-0.01
7	0.07	0.07	0.01	0.01	0.01	0.01	0

Slope Estimate Percent Change

	1	2	3	4	5	6	7
1	0	-0.02	0.99	0.99	0.99	0.99	1.00
2	0.020	0	0.99	0.99	0.99	0.99	1.00
3	-171.23	-174.72	0	0.00	0.03	0.01	0.10
4	-171.23	-174.72	0.00	0	0.03	0.01	0.10
5	-175.85	-179.44	-0.03	-0.03	0	-0.02	0.07
6	-172.59	-176.10	-0.01	-0.01	0.02	0	0.09
7	-189.89	-193.76	-0.11	-0.11	-0.08	-0.10	0

Where the numerical model-mapping $\textit{model name} \mapsto \{1, 2, \ldots, 7\}$ is given in the model definition table.

Discussion

The results have shown that changes in modeling strategy have little impact on the estimate of the intercept parameter. This is supported by the fact that the maximum absolute difference between estimates for the intercept is a 7% change. This consistency also supports a stronger conlcusion that parameter estimates for intercept agree in both sign and magnitude for all models.

These statements are not true for covariate parameter estimates. While these parameters are comparable within similar model frameworks, the estimates agree on only sign across all models. This is not an unexpected result, since parameter estimates for slopes are heavily dependent on how observations are corelated, and this concept is approached differently in all of the models employed.

It should be noted that, while the estimates for the slope parameters have been estimated to be "significant", the magnitude of these parameter estimates deviates very little from 0. Given this information, it would not be unreasonable to conclude that the effect of CD19 can be almost completely ignored. Significant nested model comparisons were performed to test for the significance of the CD19 covariate. It was found that there was sufficient evidence to include the covariate when compared to the null model in most situations involving log-transformed variables, and when the subject term had already been included in as a Fixed Effect.

The standard errors of both parameter estimates (intercept and slope) indicate a trend of increased uncertainty in parameter estimates as methodologies move from incorperating subject as a fixed effect, to incorperating it as a random effect. While this result was not initial expected, it is plausible in light of the nested model comparison results, and would indicate that the subject effect is likely correlated with CD19. Which would imply that the random effect assumption: "unobserved. . . [individual/subject] heterogeneity is uncorrelated with idependent variables" (Wooldridge 2010) is likely violated.

Limitations and Future Research

Initial quality control measures for the scRNAseq data found an extremely high presence of mitochondrial RNA (mRNA). Mitochondrial functionality is essential to cellular processeses, upon death such functions cease, and mitochondria degrade. As a result, mRNA content is used as a quality control measure to indicate progression of cellular death. The Seurat tutorial for single-cell analysis recommended filter parameters of %mRNA < %5; however this threshold elimated all data when used in conjunction with the other recommended quality control measures. A final determination of %mRNA < %60 was used to preserve as much of the clustering-structure as possible, with the trade-off of analyzing dead cells. Even after altering the QC thresholds to allow for more data, observational imbalance between subjects was observed, and remained mostly unaccounted-for throughout the analysis.

The results of this investigation are based upon mostly non-living biological samples. While interpretation of model parameters is certainly possible, the individual estimates are less meaningful considering the contex than the estimate trends that we are seeking. The sign of the estimates for the slope parameter, which would be the most contextually illuminating (as there were no control-subjects in the data), demonstrate that increased values of CD19 are marginally (if at all) associated with higher values of MALAT1. However, as previously indicated, the magnitude of these estimates reinforce interpretational futility.

The analysis conducted in this investigation is inadequate and inaccurate. The analysis conducted here is inaccurate because of biologically poor quality data, and imbalance between subject information. Additionally, the model estimates being compared can not be used for inference on the source of estimate variances because of multi-factor changes across each estimate (model fitting technique, linear predictor changes, data transformation differences). The results of this investigation are inadequate because the original goal of the investigation was to investigate models over single-cell data and isolate the specific effects that the modeling strategies have over parameter estimates. Since only vauge conlcusions regarding slope and intercept were obtained, this investigation has fallen short of its objective.

Consequently, the future of research for this project is very broad in scope. Immediate considerations will be made to compare more simplistic modeling strategies over more similar models.

References

Arazi, Arnon, Deepak A Rao, Celine C Berthier, Anne Davidson, Yanyan Liu, Paul J Hoover, Adam Chicoine, et al. 2018. "The Immune Cell Landscape in Kidneys of Lupus Nephritis Patients." *bioRxiv*. Cold Spring Harbor Laboratory, 363051.

"CD19 Gene - Genecards | Cd19 Protein | Cd19 Antibody." n.d. `https://www.genecards.org/cgi-bin/carddisp.pl?gene=CD19&keywords=CD19`.

"Gene Expression Profiling - Wikipedia." n.d. `https://en.wikipedia.org/wiki/Gene_expression_profiling`.

"MALAT1 Gene - Genecards | Malat1 Rna Gene." n.d. `https://www.genecards.org/cgi-bin/carddisp.pl?gene=MALAT1`.

Satija, R, and others. 2018. "Seurat: Guided Clustering Tutorial." *Satija Lab Http://Satijalab. Org/Seurat/Pbmc3k_tutorial. Html*.

Wooldridge, Jeffrey M. 2010. *Econometric Analysis of Cross Section and Panel Data*. MIT press.

Appendix

Plots

fig 1

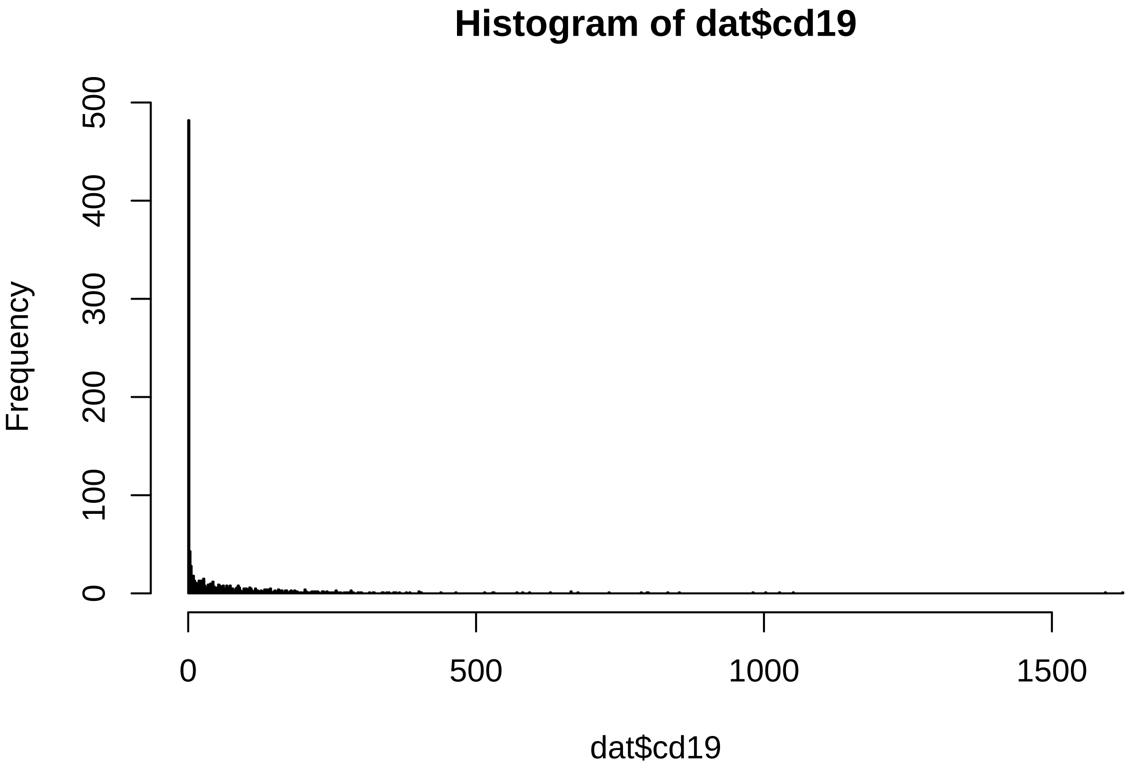


fig 2

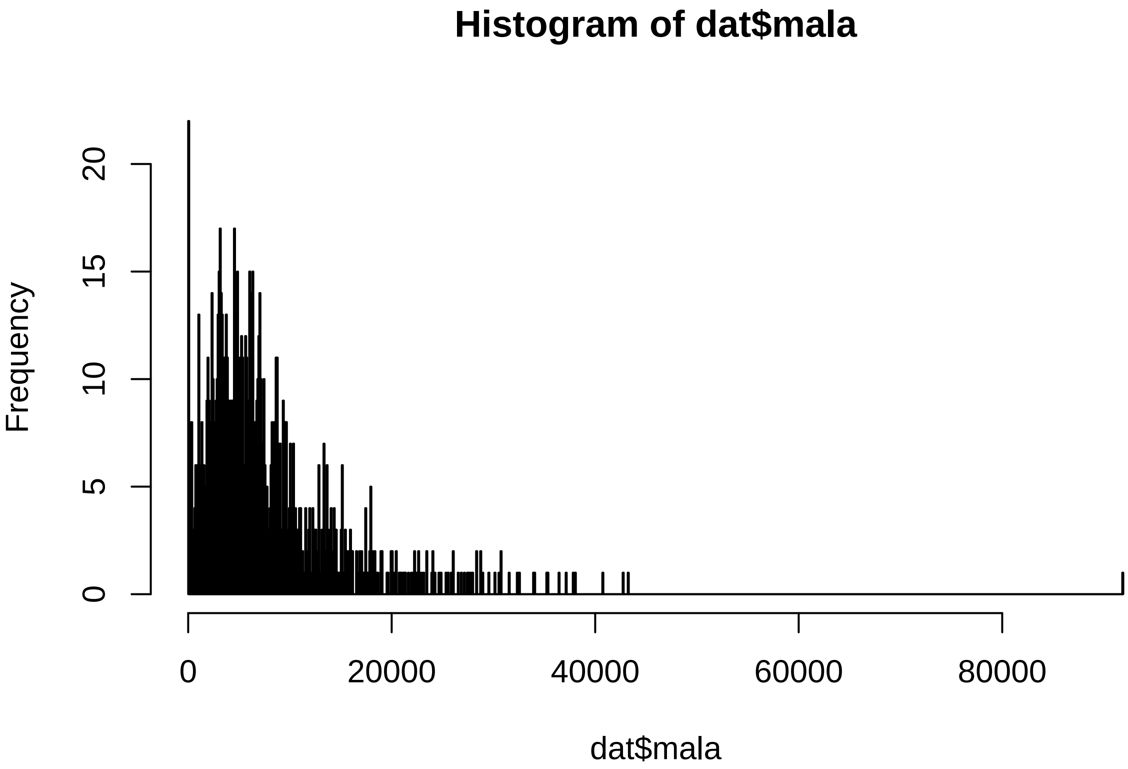

Histogram of dat$mala
Frequency
0
5
10
15
20
0
20000
40000
60000
80000
dat$mala

fig 3

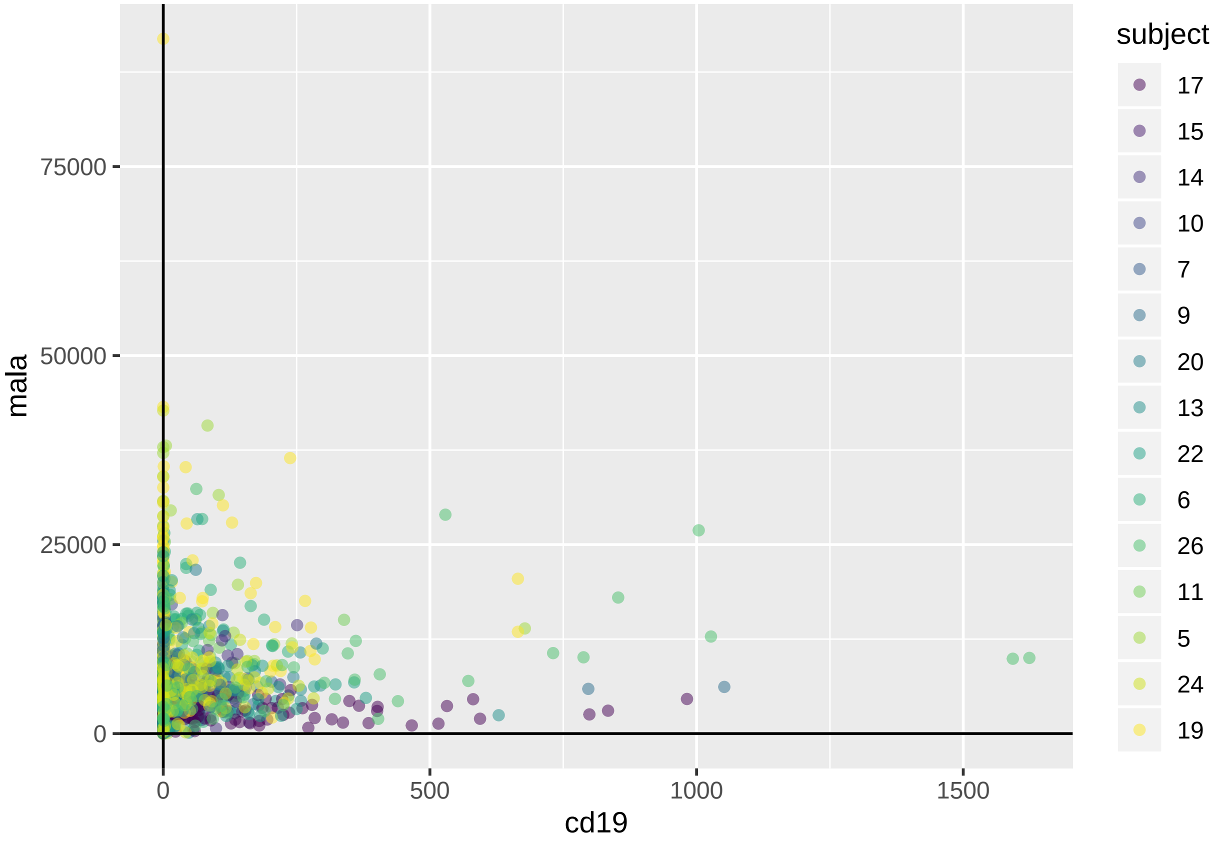

mala
75000
50000
25000
0
0
500
1000
1500
cd19
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19

fig 4

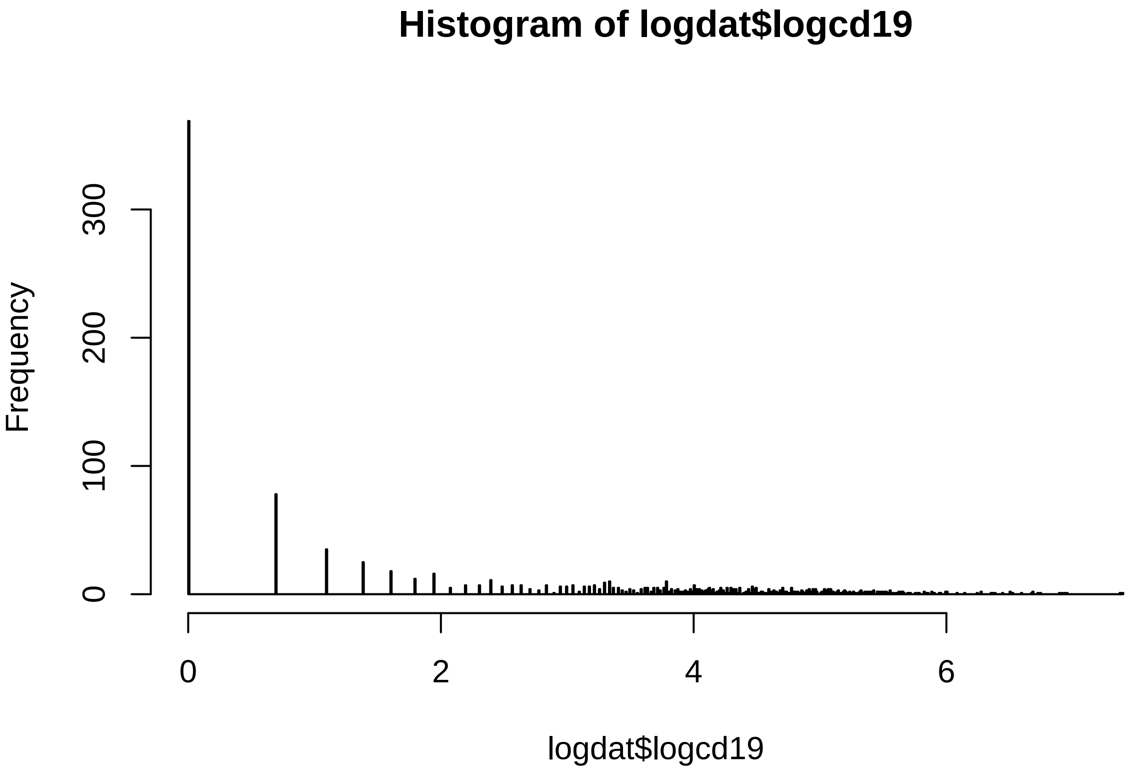

Histogram of logdat$logcd19
Frequency
0
100
200
300
0
2
4
6
logdat$logcd19

fig 5

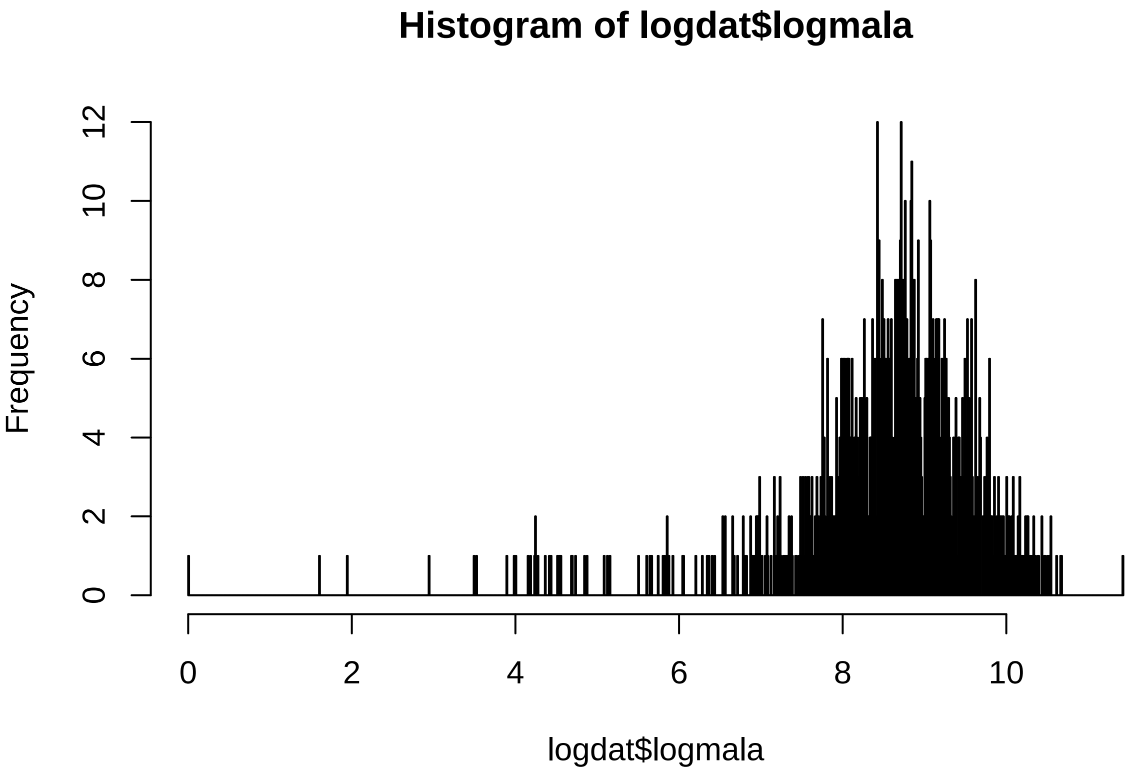
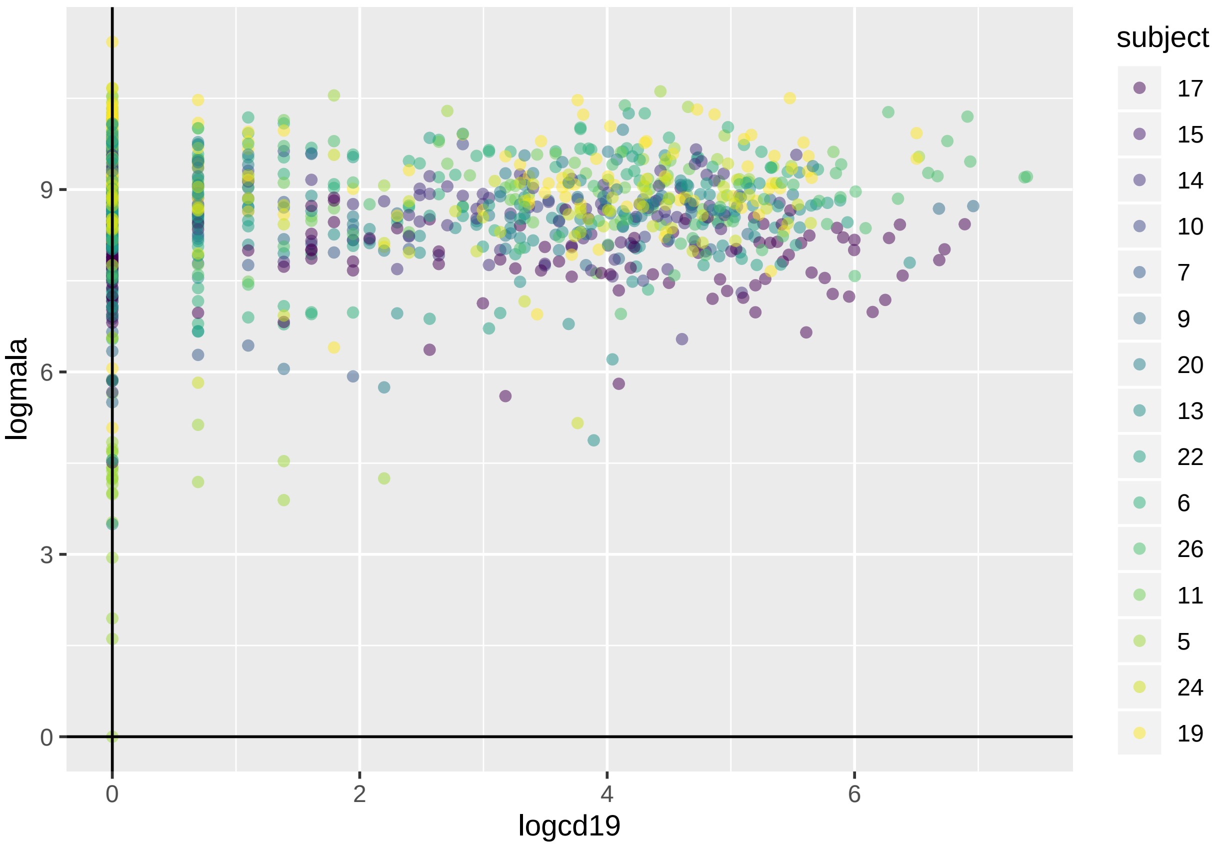
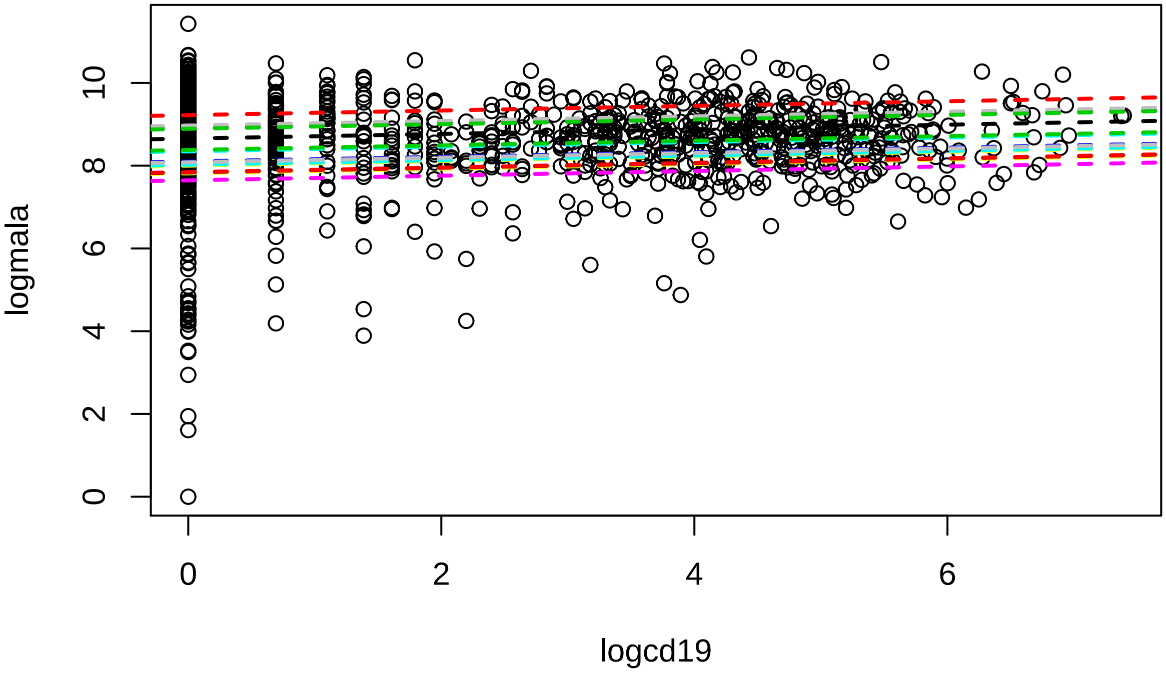
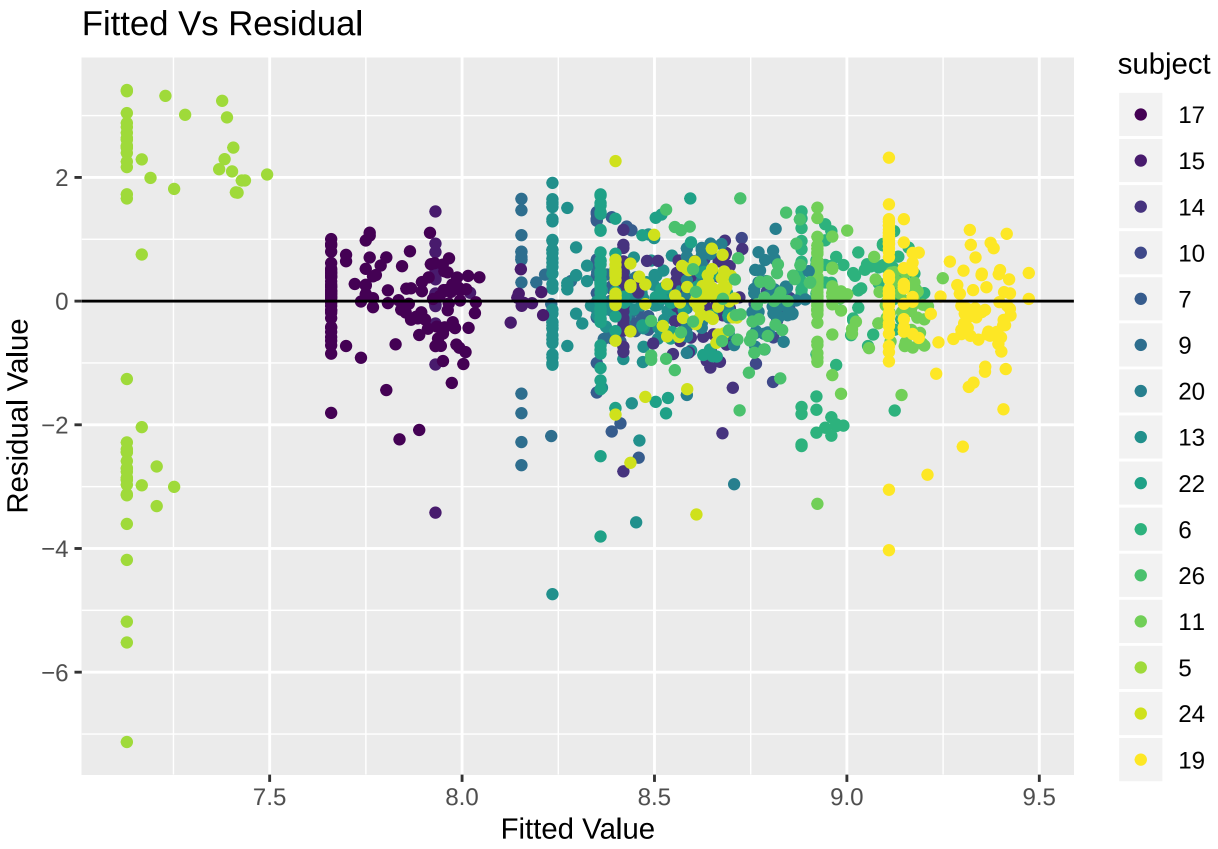

Histogram of logdat$logmala
Frequency
0
2
4
6
8
10
12
0
2
4
6
8
10
logdat$logmala

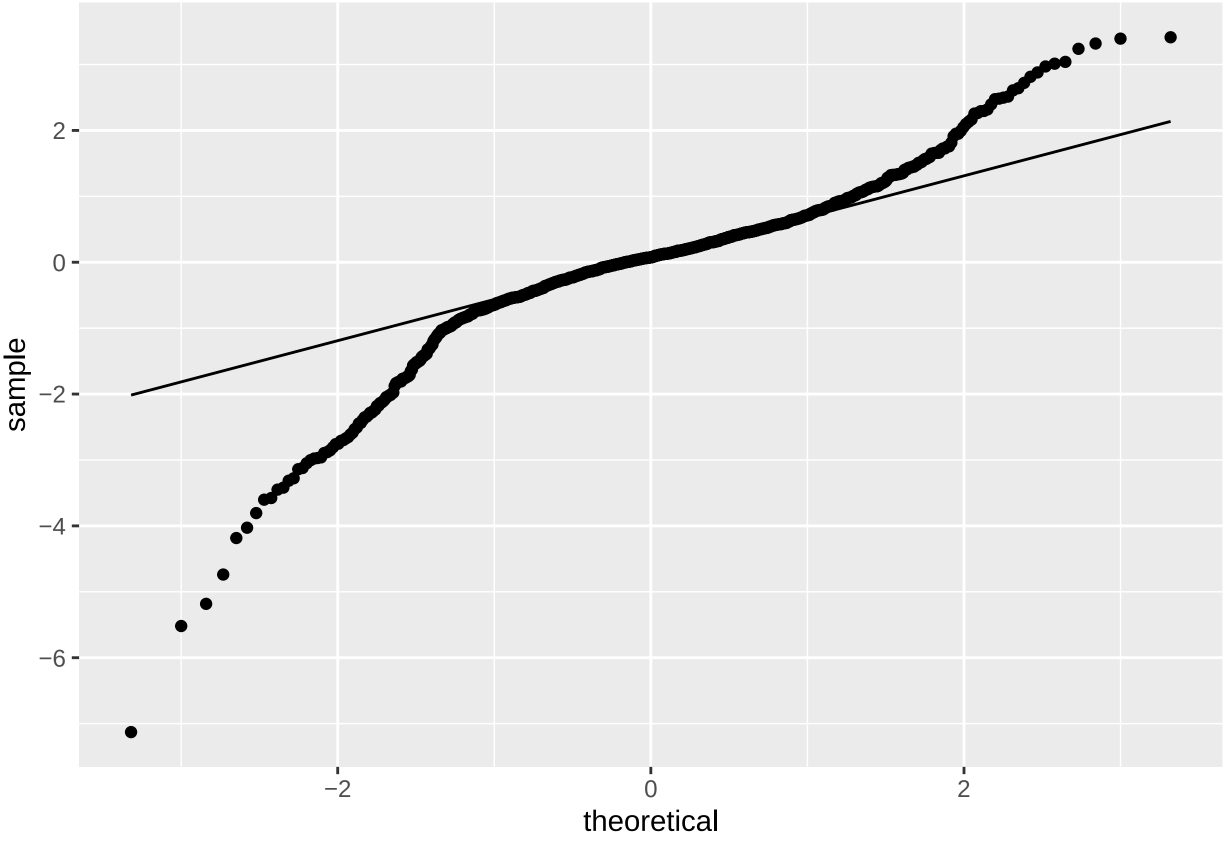
fig 6



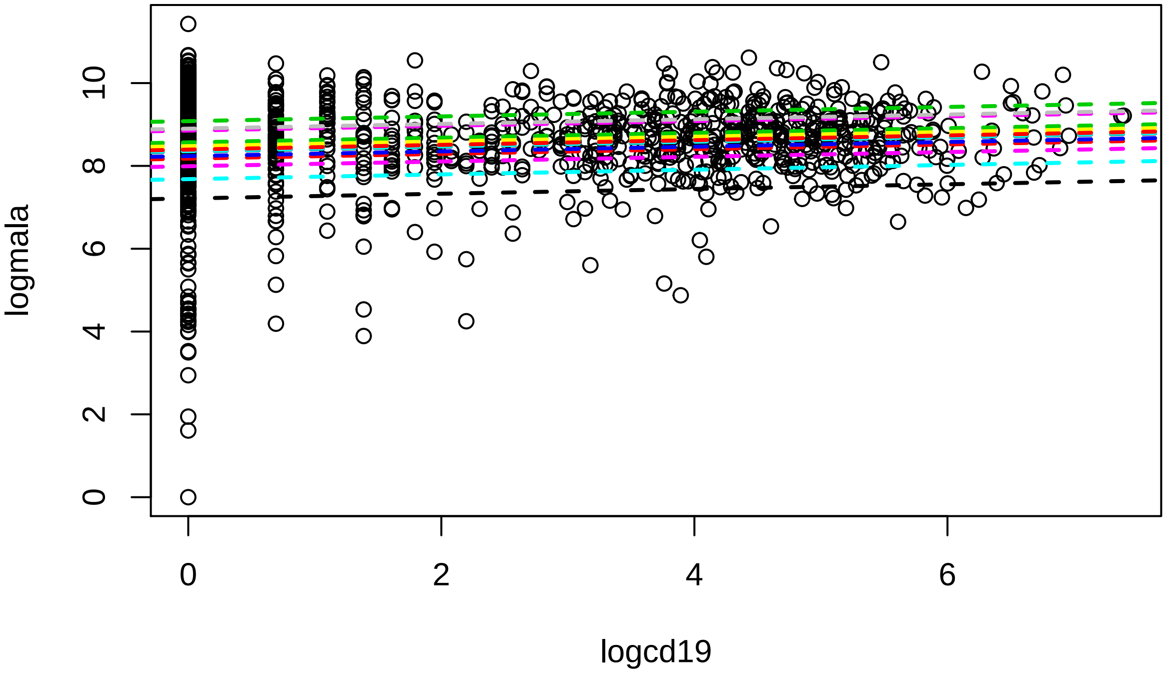
LMwFE Plots

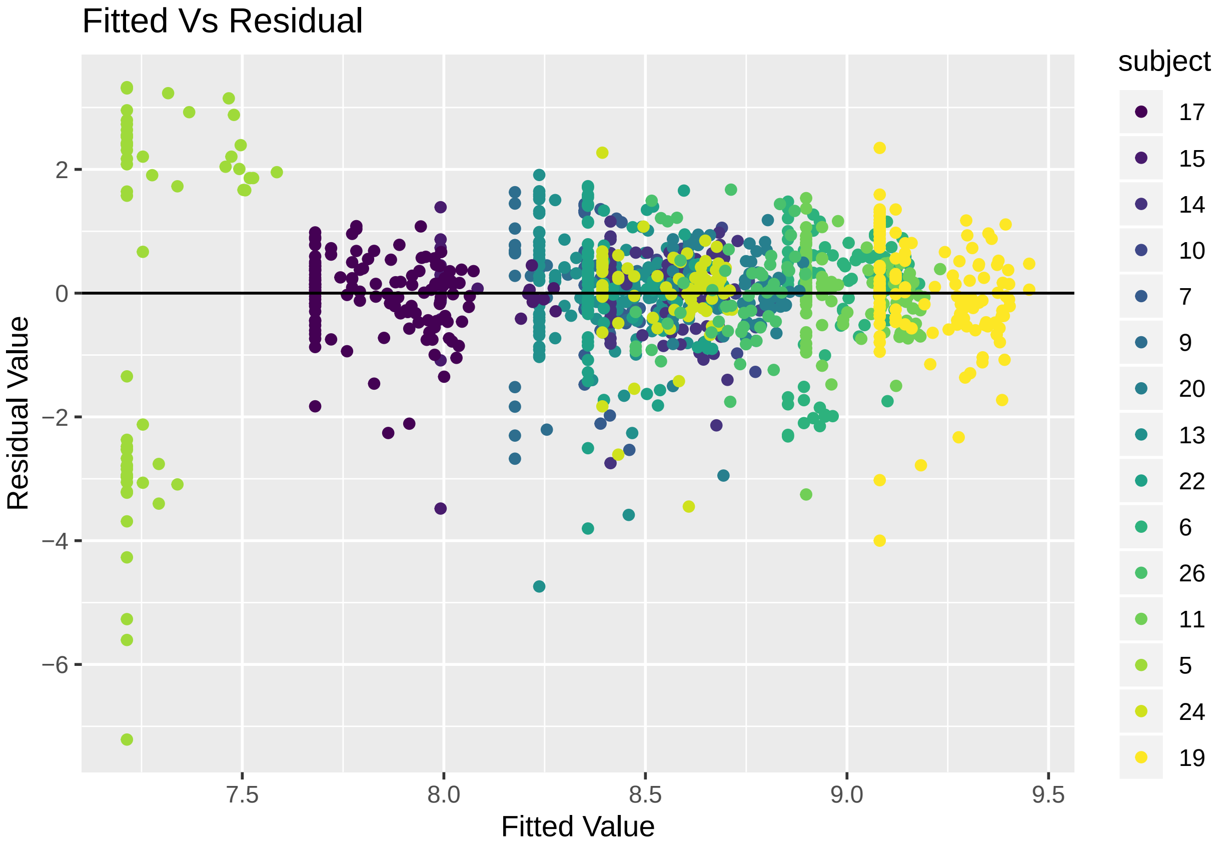


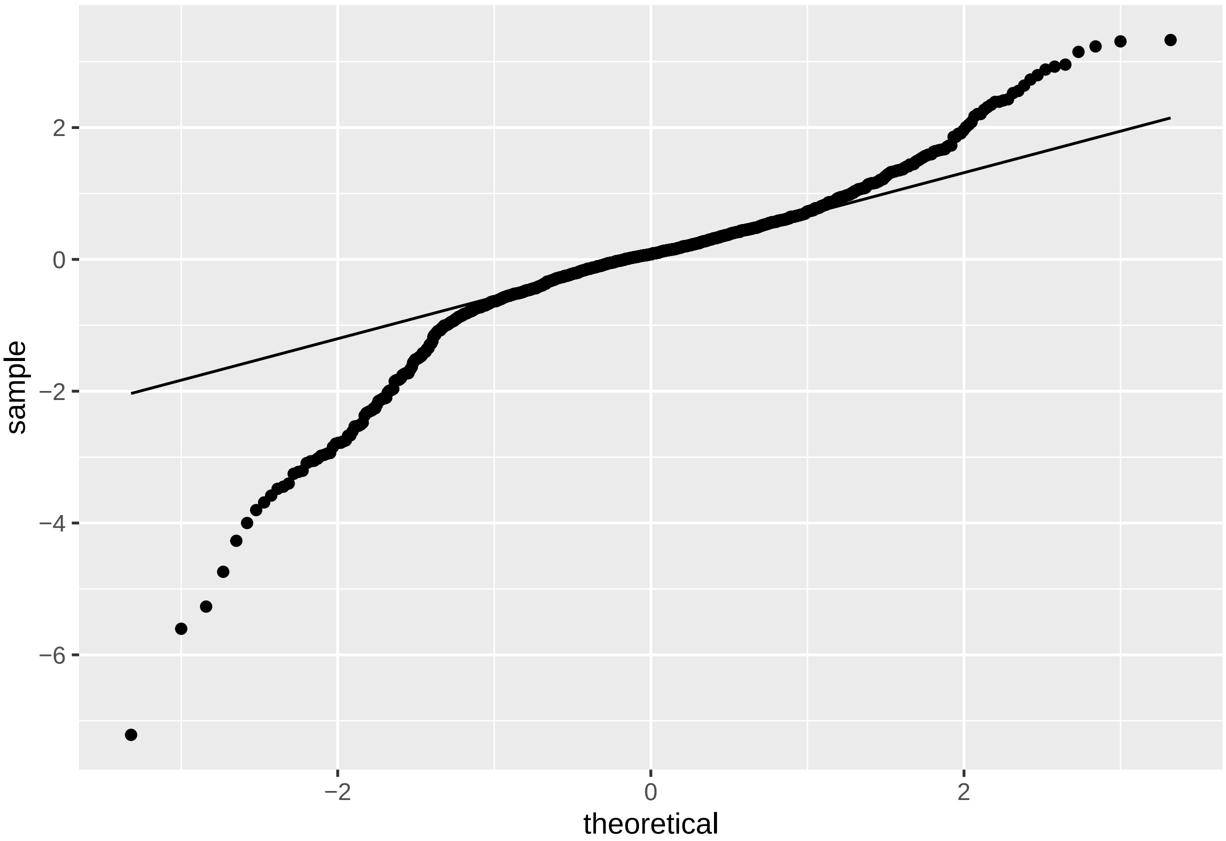

Fitted Vs Residual
Residual Value
2
0
−2
−4
−6
7.5
8.0
8.5
9.0
9.5
Fitted Value
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19


sample
2
0
−2
−4
−6
−2
0
2
theoretical

LMMwRE Plots

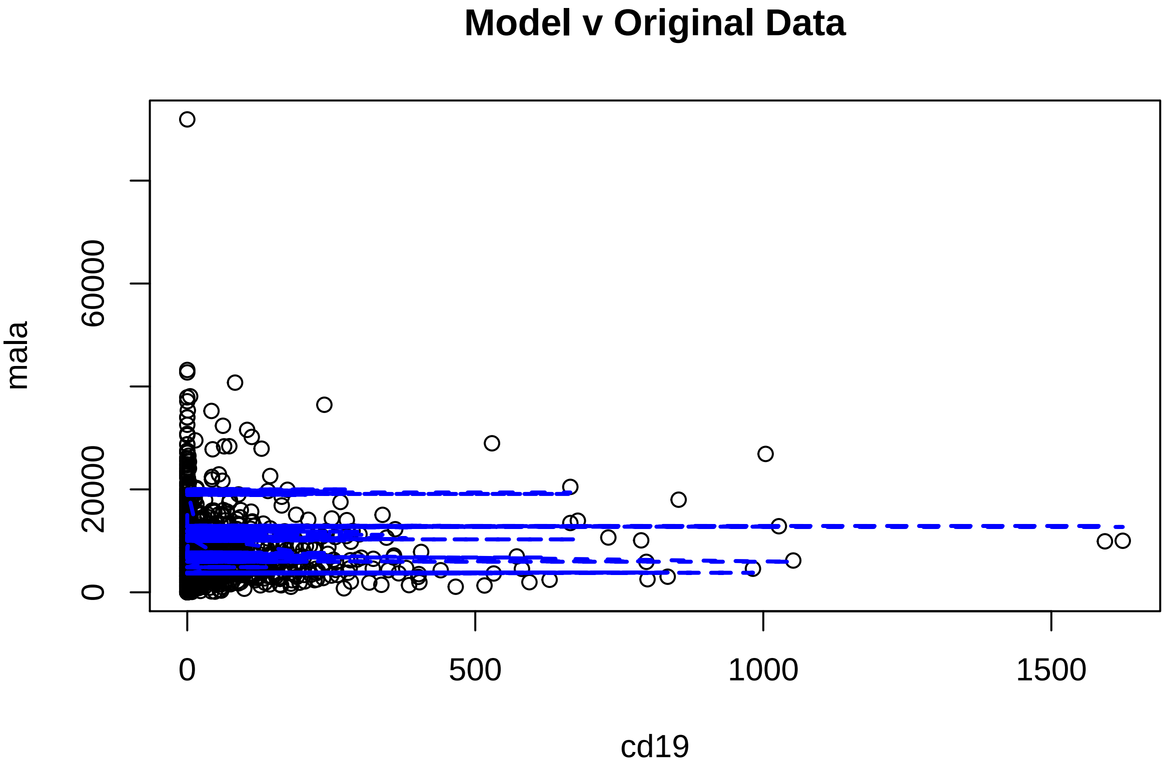


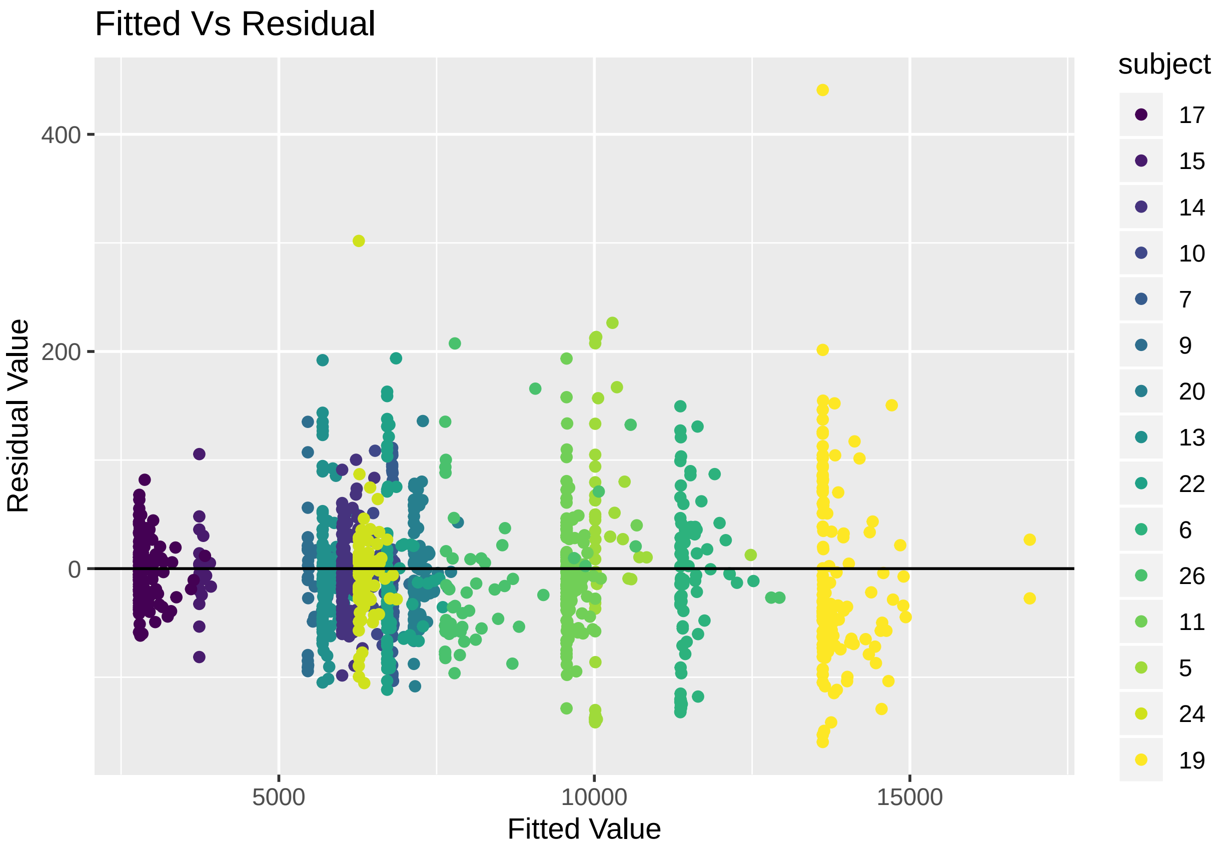


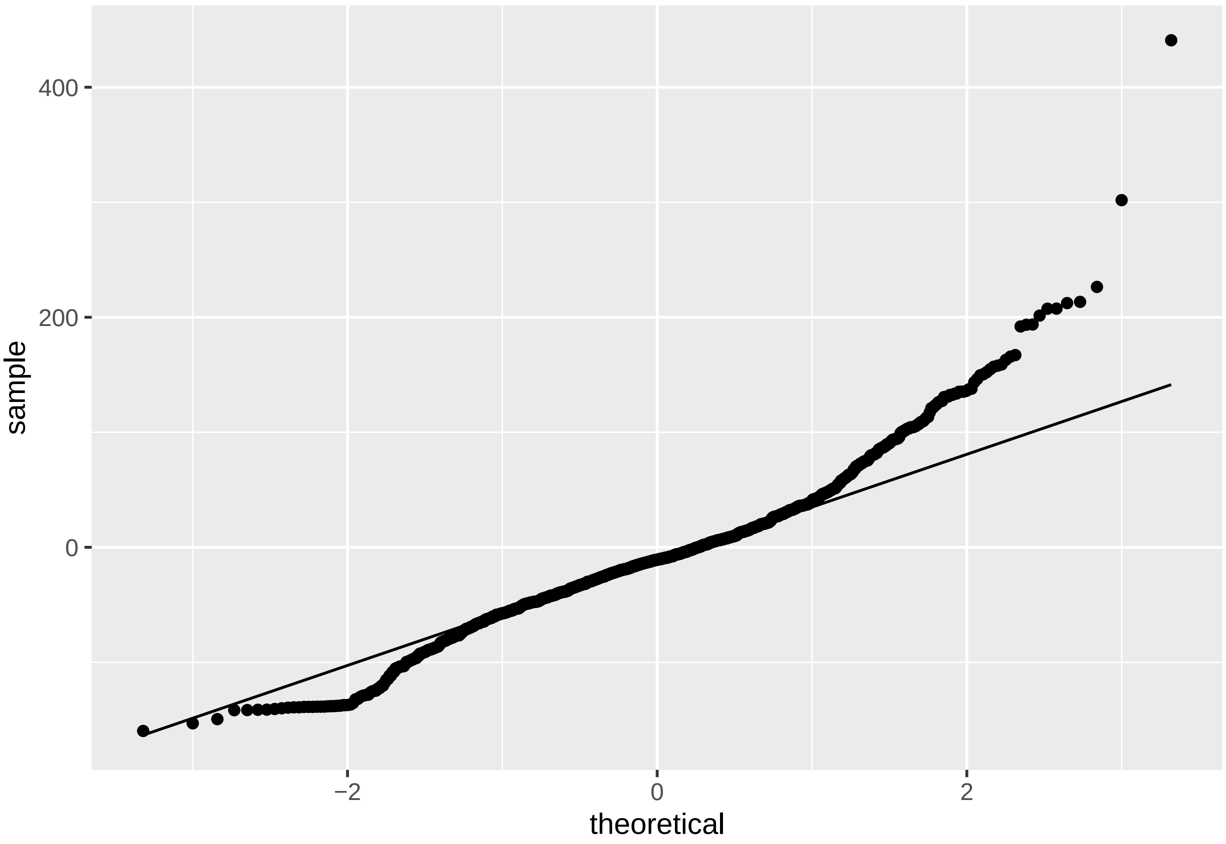


POI Plots

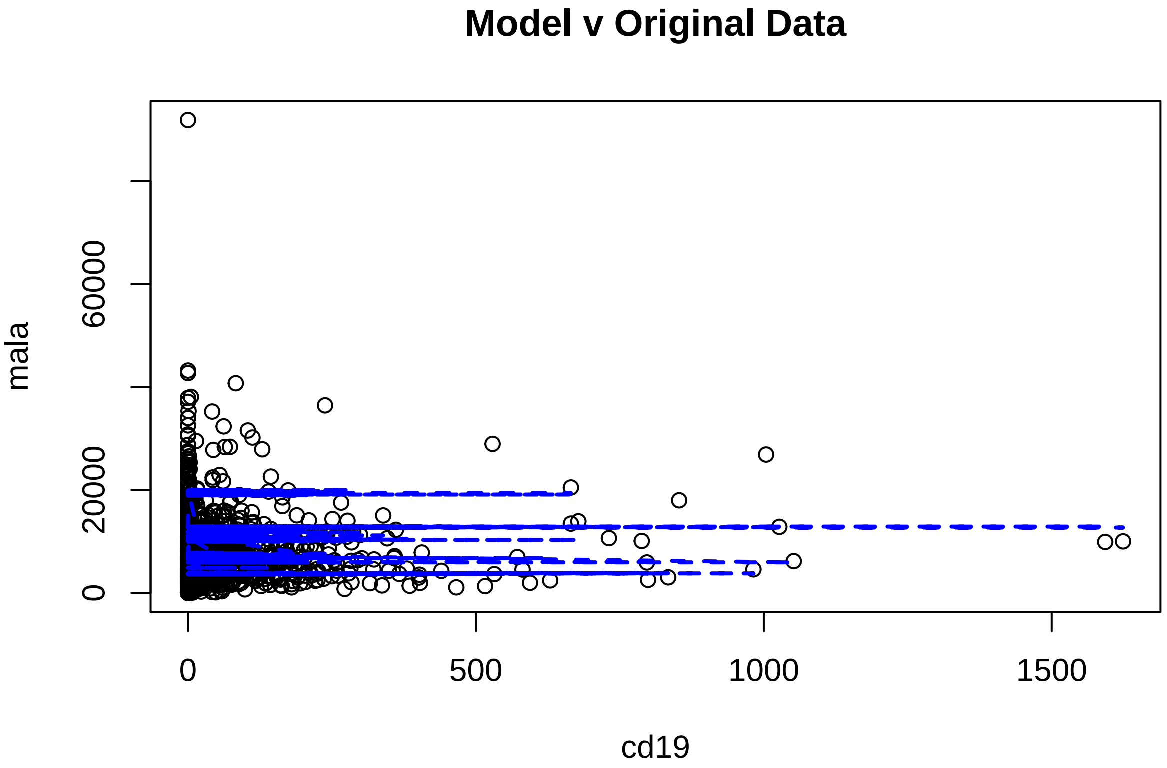
```
## [1]    0 1624
```

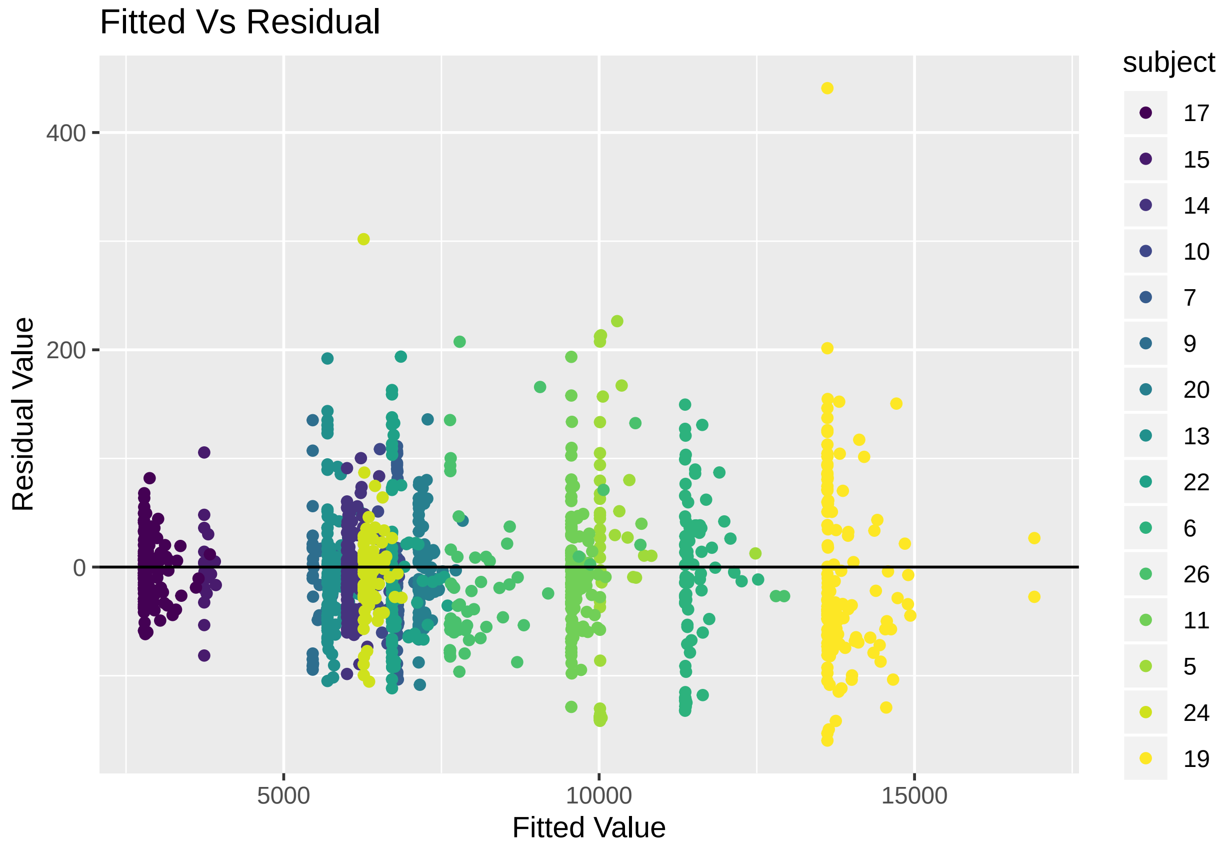


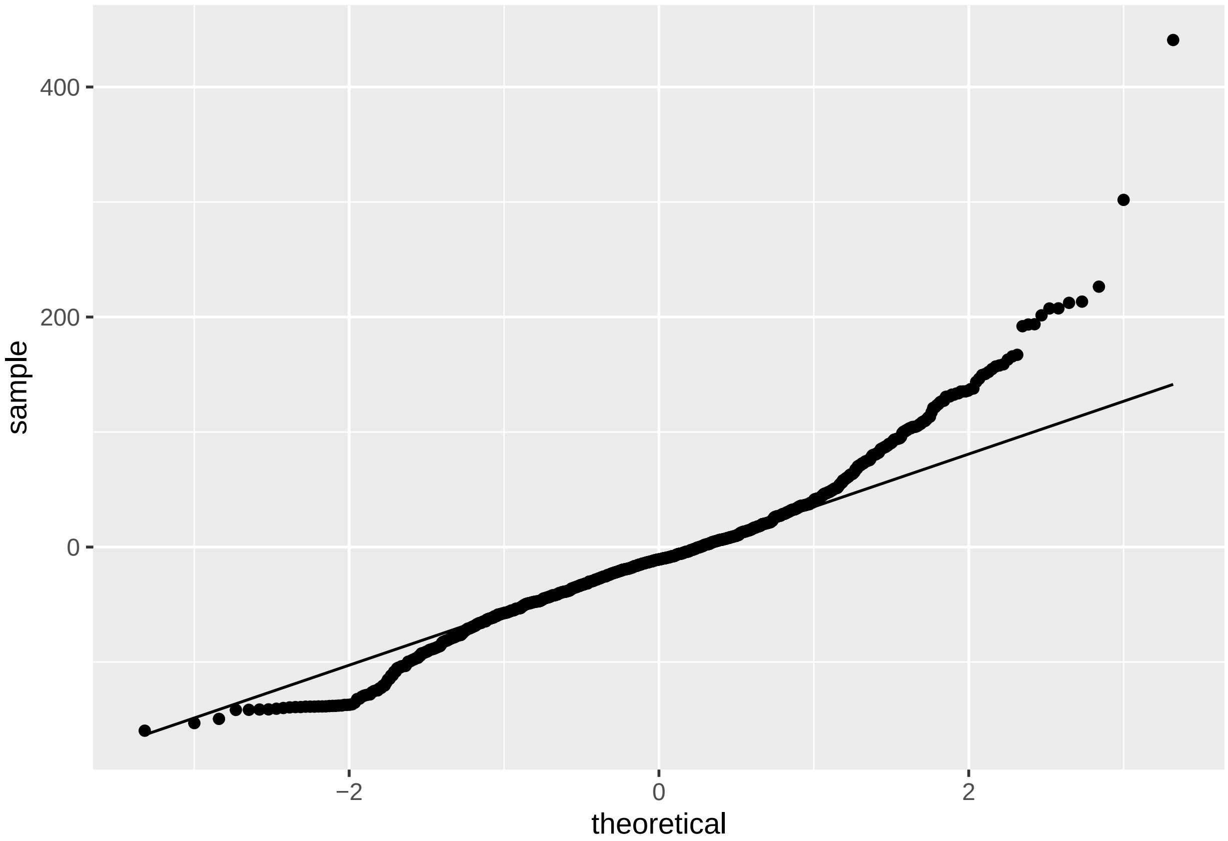

Fitted Vs Residual
Residual Value
400
200
0
5000
10000
15000
Fitted Value
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19


sample
400
200
0
−2
0
2
theoretical

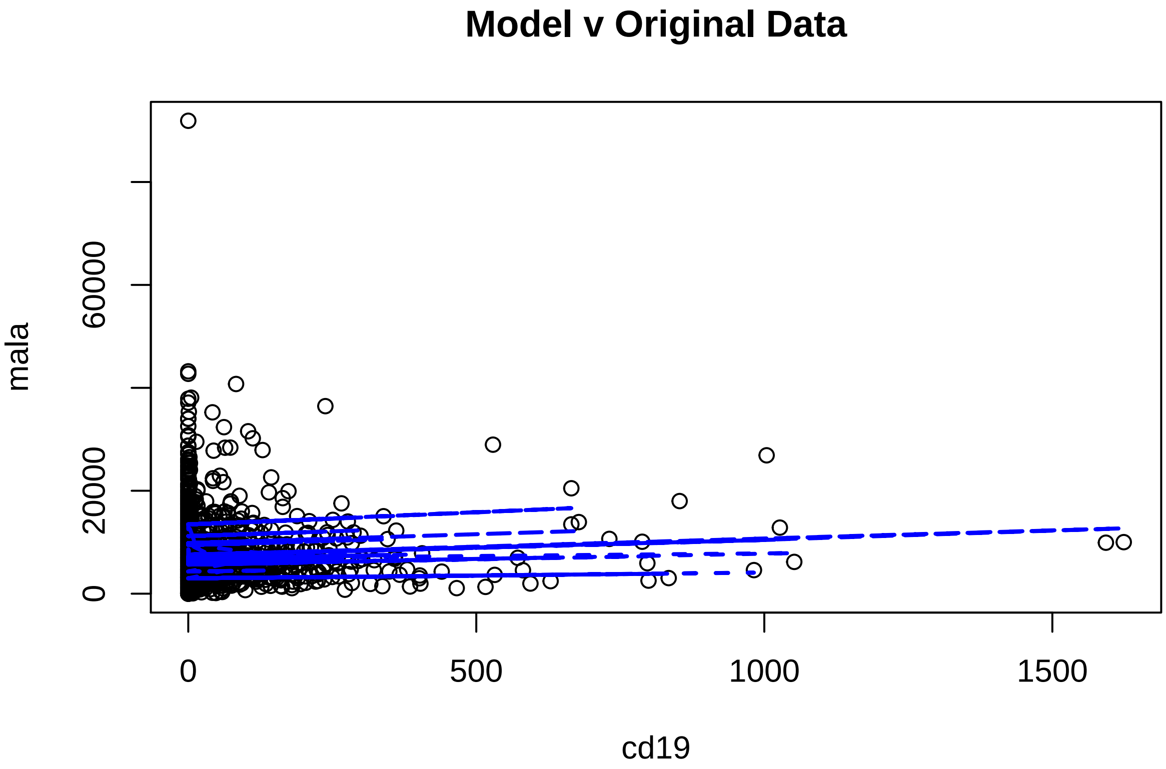
POIql Plots

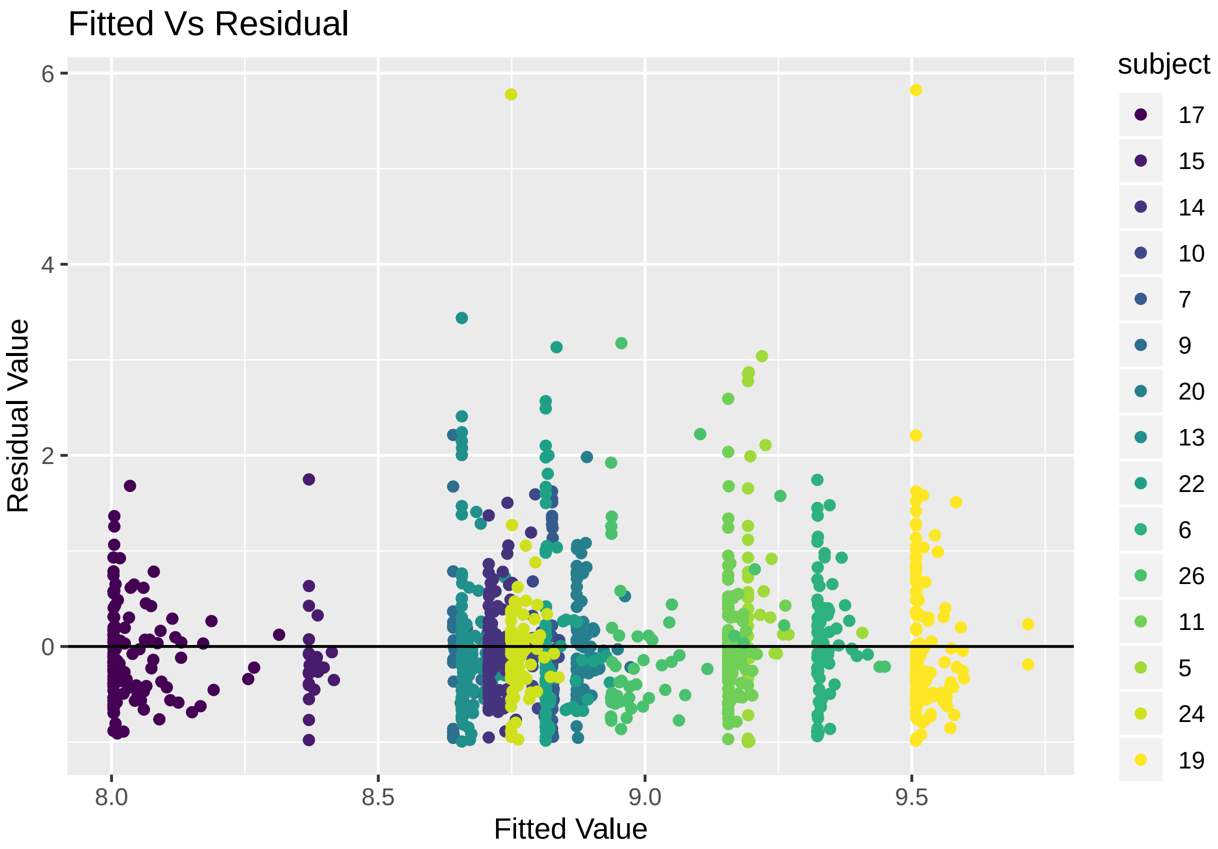

Model v Original Data
mala
0
20000
60000
cd19
0
500
1000
1500

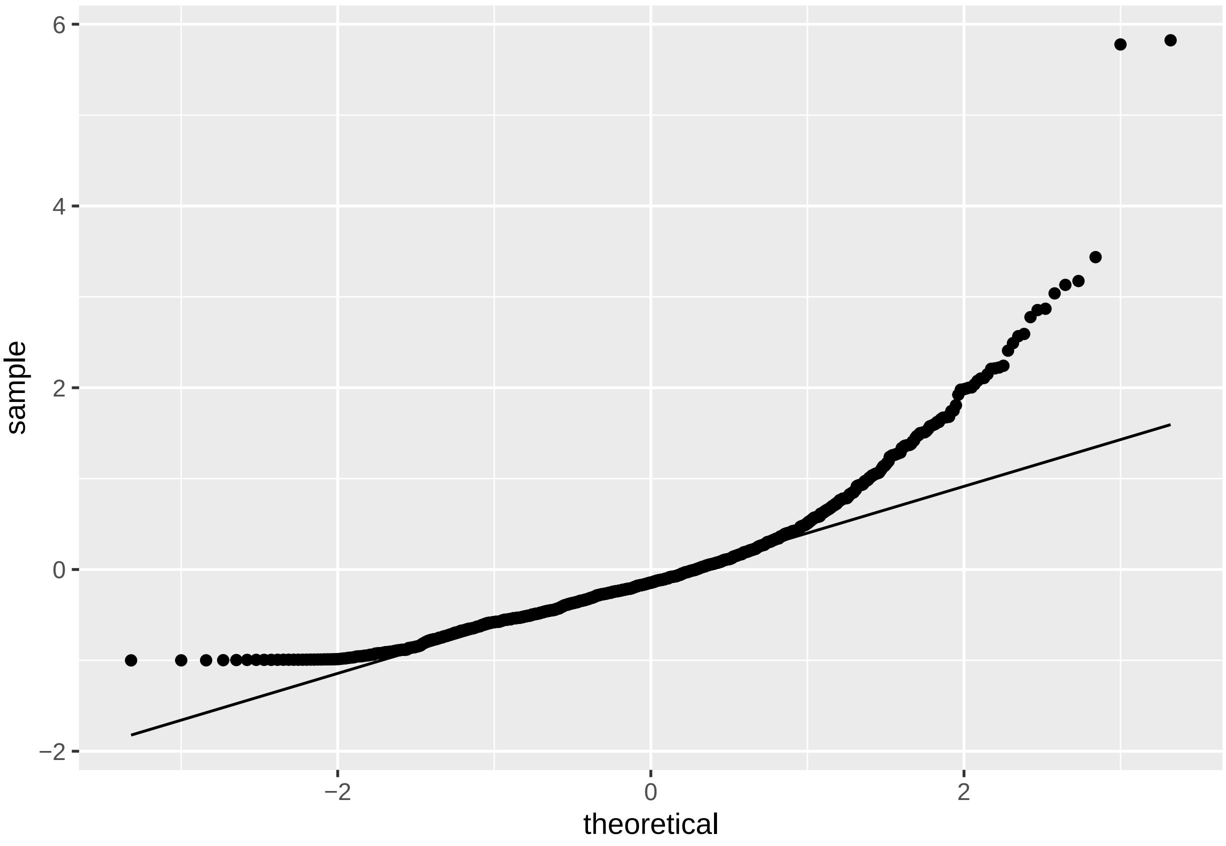

Fitted Vs Residual
Residual Value
0
200
400
Fitted Value
5000
10000
15000
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19



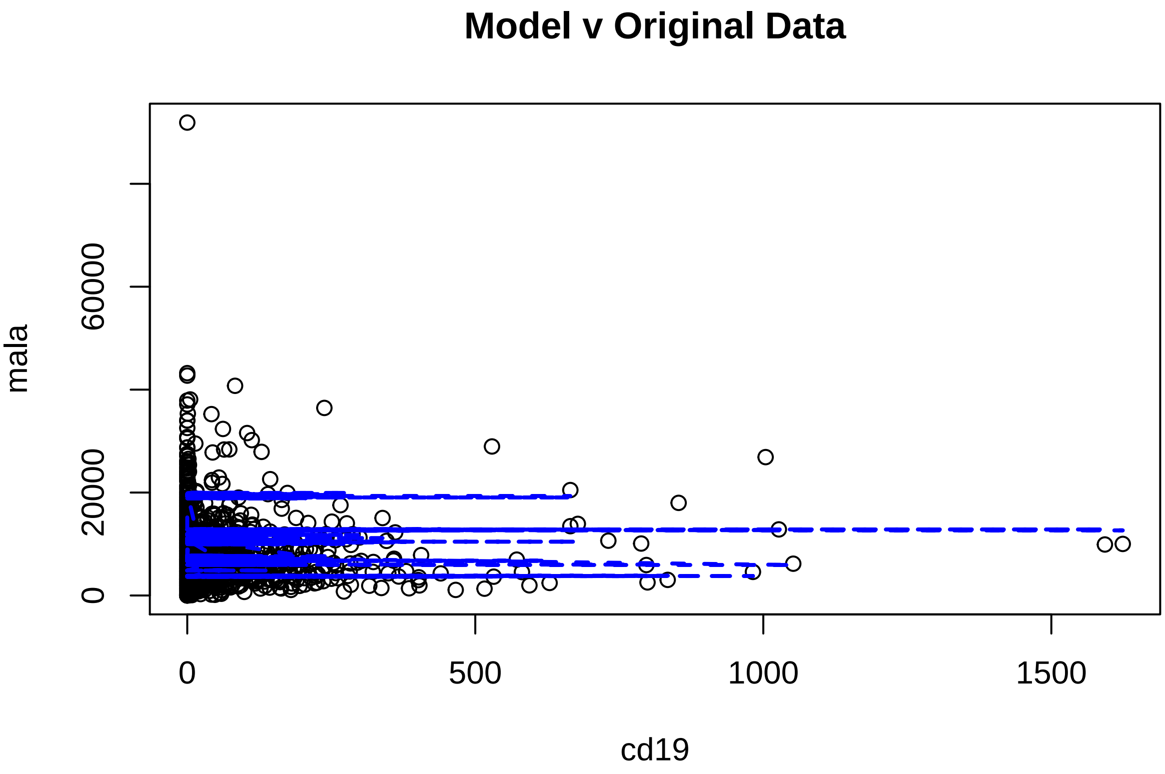
POIlmm Plots

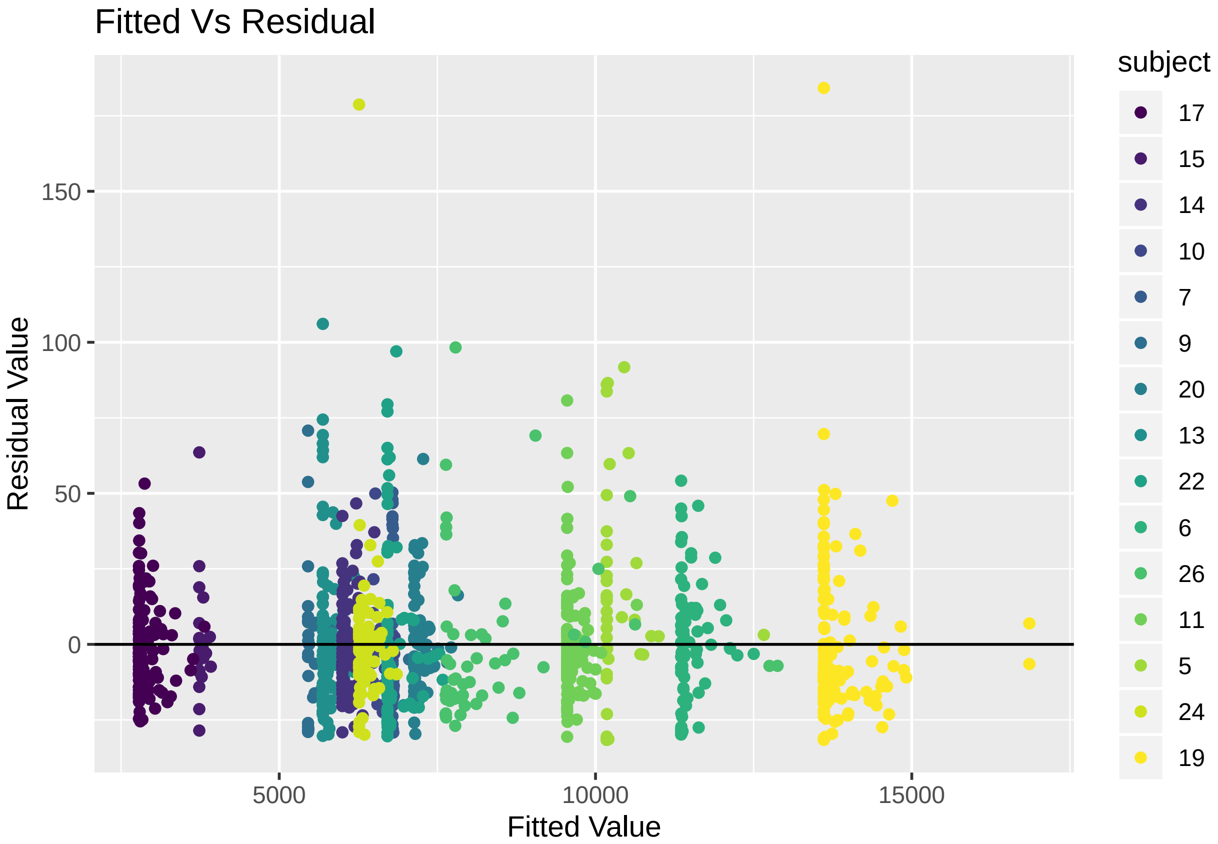


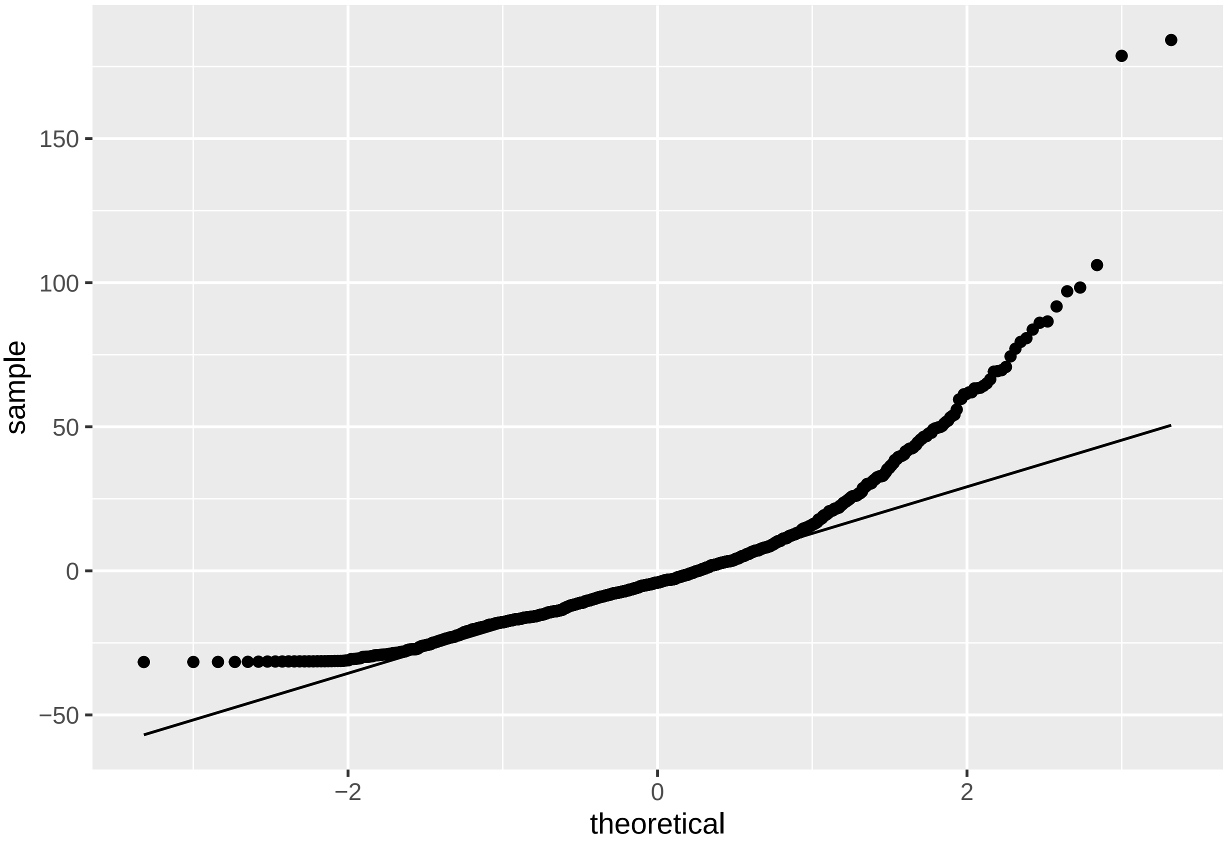

Fitted Vs Residual
Residual Value
Fitted Value
6
4
2
0
8.0
8.5
9.0
9.5
subject
● 17
● 15
● 14
● 10
● 7
● 9
● 20
● 13
● 22
● 6
● 26
● 11
● 5
● 24
● 19


sample
theoretical
6
4
2
0
−2
−2
0
2

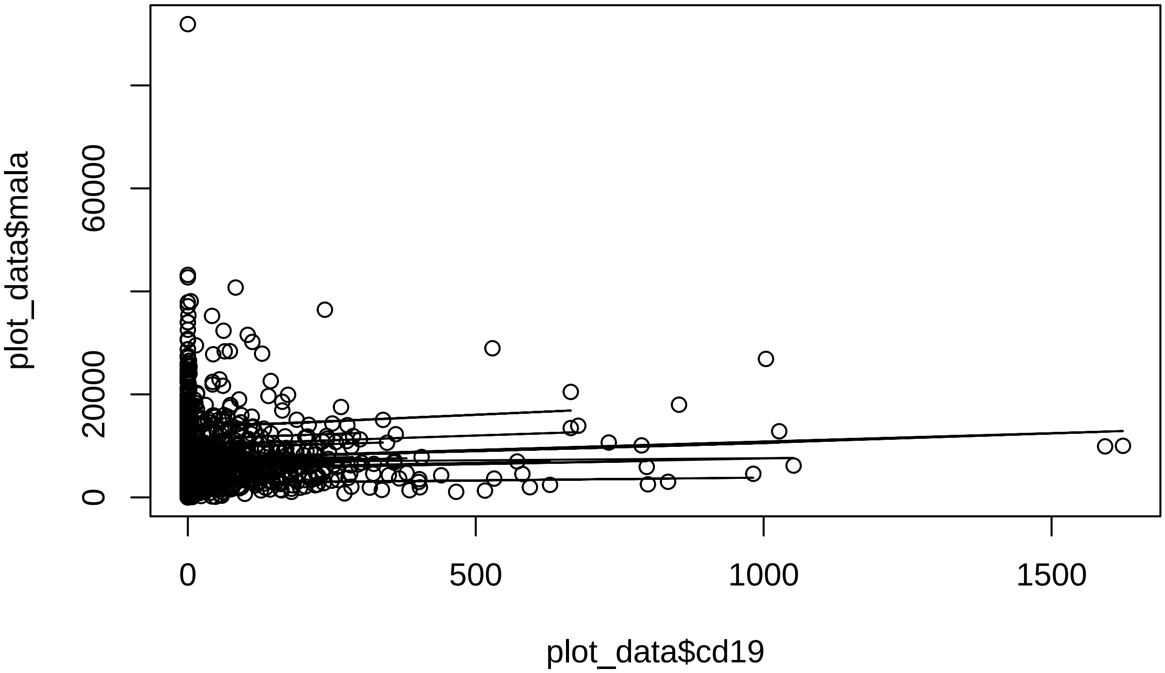
ZIPfe Plots

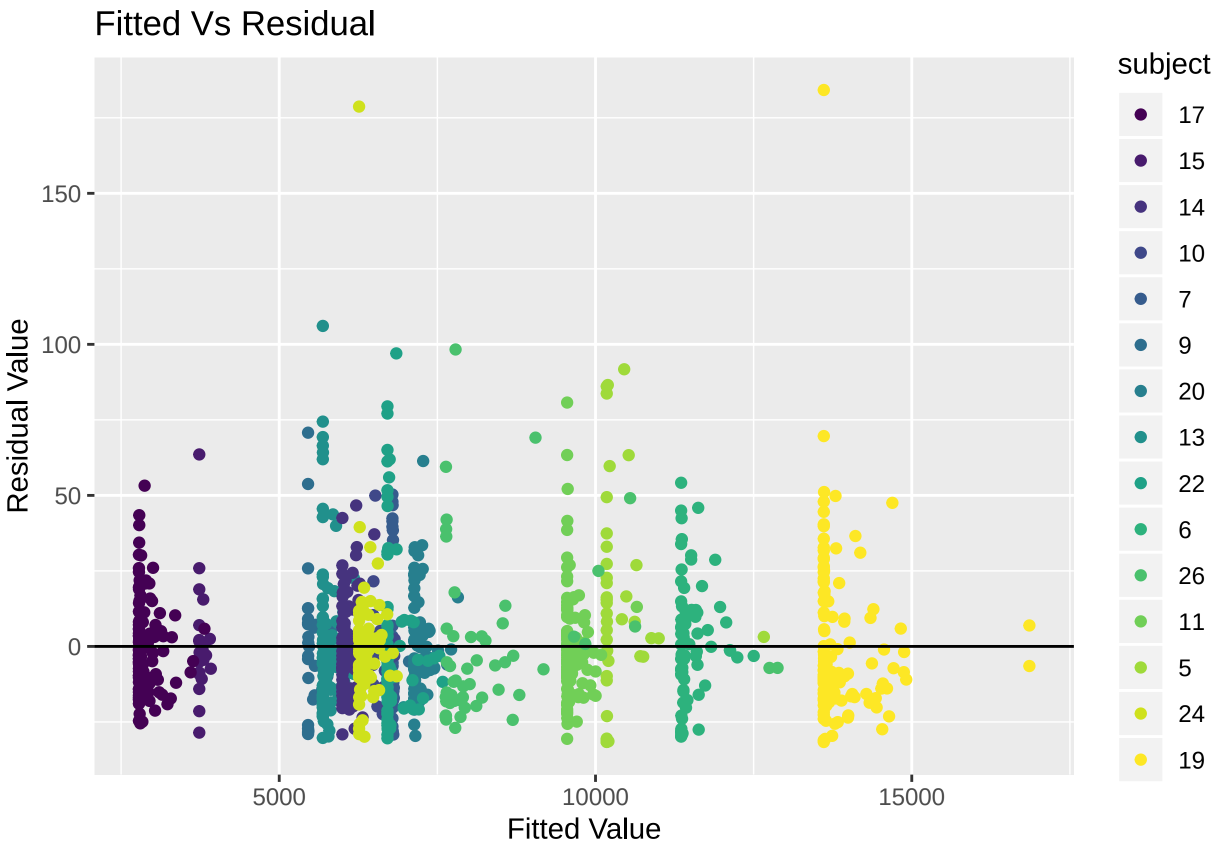

Model v Original Data
mala
0
20000
60000
0
500
1000
1500
cd19

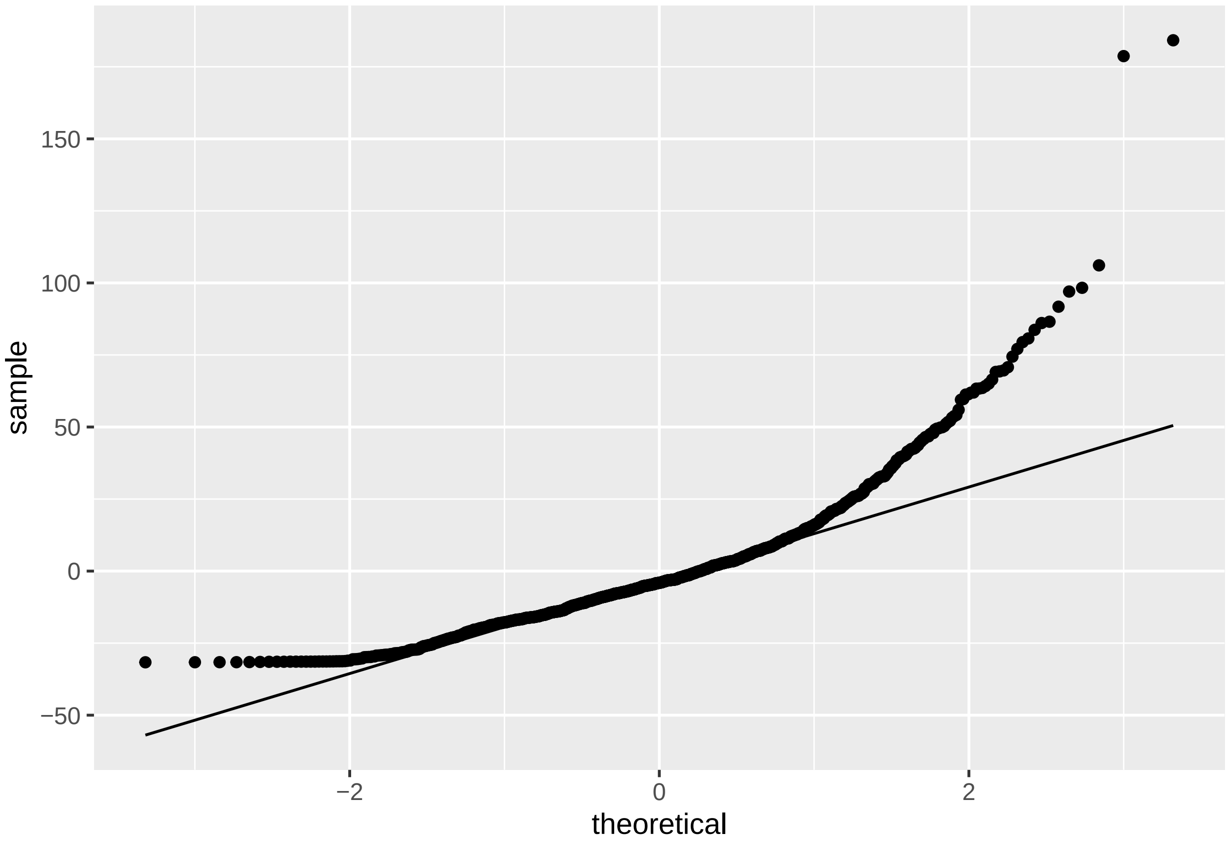

Fitted Vs Residual
Residual Value
0
50
100
150
5000
10000
15000
Fitted Value
subject
● 17
● 15
● 14
● 10
● 7
● 9
● 20
● 13
● 22
● 6
● 26
● 11
● 5
● 24
● 19


sample
150
100
50
0
−50
−2
0
2
theoretical

ZIPre Plots


plot_data$mala
0
20000
60000
0
500
1000
1500
plot_data$cd19


Fitted Vs Residual
Residual Value
150
100
50
0
5000
10000
15000
Fitted Value
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19


sample
150
100
50
0
−50
−2
0
2
theoretical

Code

Please note that the code displayed here is only a fraction of the code used for this project. The code below does display all outcomes referenced within the paper (with exception of nested model comparisons).

If there are results in the paper that I have not accounted for in the displayed code, you may referrence the following GitHub repository, where all project code is contained.

`https://github.com/leepanter/BIOS6643FinalProject`

There is a ReadMe which contains a list of relevant scripts, and a description of that script.

Also, please feel free to reach out to me if you have questions.

```r
####    Final Project Script     ####
####    Script Name: FinalProjectScript.R

#-------------------------------------------------------------------------#
####    Description:     ####




####    Script Dependencies  ####

# Package Dependencies:
library(lme4)
library(ggplot2)
library(nlme)
library(MASS)
library(pscl)
library(stargazer)
library(GLMMadaptive)
library(lattice)
library(lmerTest)
library(ggExtra)
library(downloader)
library(RCurl)

# Set Working Directory
WD="/Users/lee/Documents/GitHub/BIOS6643FinalProject/BIOS6643FinalProject"
setwd(WD)

# Data Dependencies:
load("/Users/lee/Documents/GitHub/BIOS6643FinalProject/BIOS6643FinalProject/Data/ProjDat.RData")

#Variable Dependencies:

# File Dependencies
source(file = "/Users/lee/Documents/Lee/School/CU Denver/Fall 2019/BIOS 6643/FinalProject/Modeling/Mo


#-------------------------------------------------------------------------#
####    Begin Script     ####
#-------------------------------------------------------------------------#
dat$subject=as.factor(dat$subject)
```

```
subject=dat$subject

dat$malaOld=dat$mala
dat$mala=dat$mala-67
mala=dat$mala
cd19=dat$cd19

logcd19=log(dat$cd19+1, base = exp(1))
logmala=log(dat$mala+1, base = exp(1))


dat=groupedData(mala~cd19|subject, data = dat)

logdat=data.frame(logmala, logcd19, subject)
logdat=groupedData(logmala~logcd19|subject, data = logdat)

################################################################################

####    Initial Data Summary Plots   ####

####  CD19 histogram
hist(dat$cd19, breaks=1000)

####  MALAT1 histogram
hist(dat$mala, breaks=1000)

####  MALAT1 ~ CD19 Scatter
p3=ggplot(dat, aes(x=cd19, y=mala, color=subject))+
  geom_point(alpha=0.5)+
  geom_hline(yintercept = 0)+
  geom_vline(xintercept = 0)
p3

####  logCD19 histogram
hist(logdat$logcd19, breaks=1000)

####  logMALAT1 histogram
hist(logdat$logmala, breaks=1000)

####  logMALAT1 ~ logCD19 scatter
p6=ggplot(logdat, aes(x=logcd19, y=logmala, color=subject))+
  geom_point(alpha=0.5)+
  geom_hline(yintercept = 0)+
  geom_vline(xintercept = 0)

################################################################################

####    Define LMwFE     ####
lmod.LMwFE=lm(logmala~subject+logcd19, data = logdat)
(lmod.LMwFEs=summary(lmod.LMwFE))
(AIC.lmod.LMwFE=AIC(lmod.LMwFE))
```

```
####    Model vs Original Data -- LMwFE  ####
intercept=rep(coef(lmod.LMwFE)[1], times=15)
subject.intercept=coef(lmod.LMwFE)[2:15]
subject.intercept=c(0, subject.intercept)
intercept=intercept+subject.intercept

slope=coef(lmod.LMwFE)[16]

aCoefs=intercept
bCoefs=rep(slope, times=15)

plot(logmala~logcd19)
for(i in 1:15)
{
  abline(a=aCoefs[i], b=bCoefs[i], col=(20+i), lty=2, lwd=2)
}


####    Fitted vs Residuals -- LMwFE     ####
dat$fitted=fitted(lmod.LMwFE)
dat$resid=resid(lmod.LMwFE)

p=ggplot(dat, aes(x=fitted, y=resid, color=subject))+
  geom_point()+
  geom_hline(yintercept = 0)+
  ggtitle("Fitted Vs Residual")+
  xlab("Fitted Value")+
  ylab("Residual Value")
p


####    Q-Q plot -- LMwFE    ####
q=ggplot(dat, aes(sample = resid))+
  stat_qq()+
  stat_qq_line()
q

################################################################################

####    Define LMMwRE    ####
lmod.LMMwRE=lme4::lmer(logmala~logcd19+(1|subject),
                            data=logdat, REML = T)
(lmod.LMMwREs=summary(lmod.LMMwRE))
(AIC.lmod.LMMwRE=AIC(lmod.LMMwRE))

####    Model vs Original Data -- LMMwRE     ####
intercept=rep(lme4::fixef(lmod.LMMwRE)[1], times=15)
subject.intercept=ranef(lmod.LMMwRE)$subj$'(Intercept)'
intercept=intercept+subject.intercept

slope=fixef(lmod.LMMwRE)[2]
```

```
aCoefs=intercept
bCoefs=rep(slope, times=15)

plot(logmala~logcd19)
for(i in 1:15)
{
  abline(a=aCoefs[i], b=bCoefs[i], col=(20+i), lty=2, lwd=2)
}


####    Fitted vs Residuals -- LMMwRE    ####
dat$fitted=fitted(lmod.LMMwRE)
dat$resid=resid(lmod.LMMwRE)

p=ggplot(dat, aes(x=fitted, y=resid, color=subject))+
  geom_point()+
  geom_hline(yintercept = 0)+
  ggtitle("Fitted Vs Residual")+
  xlab("Fitted Value")+
  ylab("Residual Value")
p


####    Q-Q plot -- LMMwRE   ####
q=ggplot(dat, aes(sample = resid))+
  stat_qq()+
  stat_qq_line()
q


################################################################################

####    Define POI   ####
lmod.POI=glm(mala~subject+cd19, data = dat, family = poisson(link = "log"))
(lmod.POIs=summary(lmod.POI))
(AIC.lmod.POI=AIC(lmod.POI))

####    Model vs Original Data ####
range(cd19)
xrange=seq(0,1624, length.out = 1110)
ypredict=predict(lmod.POI, list(cd19=xrange), type = "response")

plot(mala~cd19, data=dat, main="Model v Original Data")
lines(cd19, ypredict, col=(20), lty=2, lwd=2)

####    Fitted vs Residuals -- POI   ####
dat$fitted=fitted(lmod.POI)
dat$resid=residuals(lmod.POI)

p=ggplot(dat, aes(x=fitted, y=resid, color=subject))+
  geom_point()+
  geom_hline(yintercept = 0)+
```

```r
  ggtitle("Fitted Vs Residual")+
  xlab("Fitted Value")+
  ylab("Residual Value")
p

####    Q-Q plot -- POI  ####
q=ggplot(dat, aes(sample = resid))+
  stat_qq()+
  stat_qq_line()
q

################################################################################

####    Define POIql     ####
lmod.POIql=glm(mala~subject+cd19, data = dat, family = quasipoisson(link = "log"))
(lmod.POIqls=summary(lmod.POIql))

####    Model vs Original Data ####
xrange=seq(0,1624, length.out = 1110)
ypredict=predict(lmod.POIql, list(cd19=xrange), type = "response")

plot(mala~cd19, data=dat, main="Model v Original Data")
lines(cd19, ypredict, col=(20), lty=2, lwd=2)

####    Fitted vs Residuals -- POI   ####
dat$fitted=fitted(lmod.POIql)
dat$resid=residuals(lmod.POIql)

p=ggplot(dat, aes(x=fitted, y=resid, color=subject))+
  geom_point()+
  geom_hline(yintercept = 0)+
  ggtitle("Fitted Vs Residual")+
  xlab("Fitted Value")+
  ylab("Residual Value")
p


####    Q-Q plot -- POI  ####
q=ggplot(dat, aes(sample = resid))+
  stat_qq()+
  stat_qq_line()
q

################################################################################

####    Define POIlmm    ####
lmod.POIlmm=glmmPQL(mala~cd19,
                        random = list(~1|subject),
                        family = poisson,
                        data=dat)
(lmod.POIlmms=summary(lmod.POIlmm))
```

```
####    Model vs Original Data ####
pred=list()
for(i in 1:1)
{
  pred[[i]]=predict(lmod.POIlmm, level = i, type = "response")
}

plot(mala~cd19, data=dat, main="Model v Original Data")
for(i in 1:1)
{
  lines(cd19, pred[[i]], col=(20), lty=2, lwd=2)
}

####    Fitted vs Residuals ####
dat$fitted=fitted(lmod.POIlmm)
dat$resid=residuals(lmod.POIlmm)

p=ggplot(dat, aes(x=fitted, y=resid, color=subject))+
  geom_point()+
  geom_hline(yintercept = 0)+
  ggtitle("Fitted Vs Residual")+
  xlab("Fitted Value")+
  ylab("Residual Value")
p


####    Q-Q plot -- POIqlLMM     ####
q=ggplot(dat, aes(sample = resid))+
  stat_qq()+
  stat_qq_line()
q

################################################################################

####    Define ZIPfe     ####
lmod.ZIPfe=zeroinfl(mala~cd19+subject |1,
                    data=dat,
                    dist = "poisson")
(lmod.ZIPfes=summary(lmod.ZIPfe))

####    Model vs Original Data ####
xrange=seq(0,1624, length.out = 1110)
ypredict=predict(lmod.ZIPfe, list(cd19=xrange), type = "response")

plot(mala~cd19, data=dat, main="Model v Original Data")
lines(cd19, ypredict, col=(20), lty=2, lwd=2)

####    Fitted vs Residuals ####
dat$fitted=fitted(lmod.ZIPfe)
dat$resid=residuals(lmod.ZIPfe)

p=ggplot(dat, aes(x=fitted, y=resid, color=subject))+
  geom_point()+
  geom_hline(yintercept = 0)+
  ggtitle("Fitted Vs Residual")+
  xlab("Fitted Value")+
  ylab("Residual Value")
p


####    Q-Q plot -- POIqlLMM     ####
q=ggplot(dat, aes(sample = resid))+
  stat_qq()+
  stat_qq_line()
q


################################################################################

####    Define ZIPre     ####
lmod.ZIPre=mixed_model(mala~cd19,
                           random = ~1|subject,
                           data=dat,
                           family = zi.poisson(),
                           zi_fixed = ~ 1)
(lmod.ZIPres=summary(lmod.ZIPre))

####    Model vs Original Data   ####
nDF=dat

plot_data=effectPlotData(lmod.ZIPre, nDF)

plot(plot_data$mala ~ plot_data$cd19)
lines(plot_data$cd19, plot_data$fitted)


####    Fitted vs Residuals -- ZIP   ####
dat$fitted=plot_data$fitted
dat$resid=plot_data$resid

p=ggplot(dat, aes(x=fitted, y=resid, color=subject))+
  geom_point()+
  geom_hline(yintercept = 0)+
  ggtitle("Fitted Vs Residual")+
  xlab("Fitted Value")+
  ylab("Residual Value")
p


####    Q-Q plot -- ZIP  ####
q=ggplot(dat, aes(sample = resid))+
  stat_qq()+
  stat_qq_line()
q
```

```
################################################################################

# Calculate Perct change Matrices

PercentChange=function(a,b){
  out=(a-b)/a
  return(out)
}

VectorPC=function(vec){
  l=length(vec)
  out.mat=matrix(NA, nrow = l, ncol = l)
  for(i in 1:l){
    for(j in 1:l){
      out.mat[i,j]=PercentChange(vec[i], vec[j])
    }
  }
  return(out.mat)
}

####  Store Coefficients

# Intercept Coefficients and vector
int.LMwFE=coef(lmod.LMwFE)[1]
int.LMMwFE=fixef(lmod.LMMwRE)[1]
int.POI=coef(lmod.POI)[1]
int.POIql=coef(lmod.POIql)[1]
int.POIlmm=fixef(lmod.POIlmm)[1]
int.ZIPfe=lmod.ZIPfe[["coefficients"]][["count"]][["(Intercept)"]]
int.ZIPre=lmod.ZIPre[["coefficients"]][["(Intercept)"]]

InterceptVec=c(int.LMwFE,
               int.LMMwFE,
               int.POI,
               int.POIql,
               int.POIlmm,
               int.ZIPfe,
               int.ZIPre)
(PercentChange.InterceptVec=round(VectorPC(InterceptVec), 3))



# Slope Coefficients and vector
slope.LMwFE=coef(lmod.LMwFE)[16]
slope.LMMwFE=fixef(lmod.LMMwRE)[2]
slope.POI=coef(lmod.POI)[16]
slope.POIql=coef(lmod.POIql)[16]
slope.POIlmm=fixef(lmod.POIlmm)[2]
slope.ZIPfe=lmod.ZIPfe[["coefficients"]][["count"]][["cd19"]]
slope.ZIPre=lmod.ZIPre[["coefficients"]][["cd19"]]
```

```
SlopeVec=c(slope.LMwFE,
           slope.LMMwFE,
           slope.POI,
           slope.POIql,
           slope.POIlmm,
           slope.ZIPfe,
           slope.ZIPre)
(PercentChange.SlopeVec=round(VectorPC(SlopeVec),3))



#--------------------------------------------------------------------------#
####    End Script   ####
#--------------------------------------------------------------------------#





#--------------------------------------------------------------------------#
#####   Post-Script #####

####  Notes:

####  Compilation Errors:

####  Execution Errors:

####  Next Scripts to Consider:


#--------------------------------------------------------------------------#
```