Appendix

Plots

fig 1

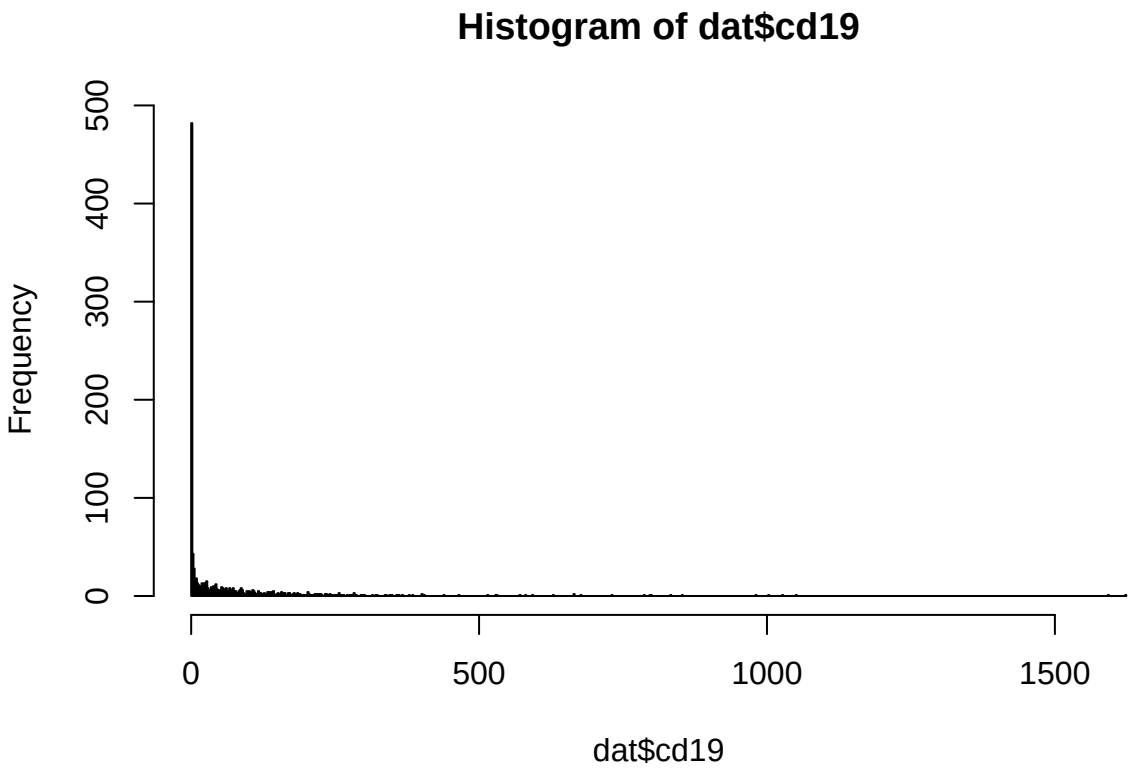


fig 2

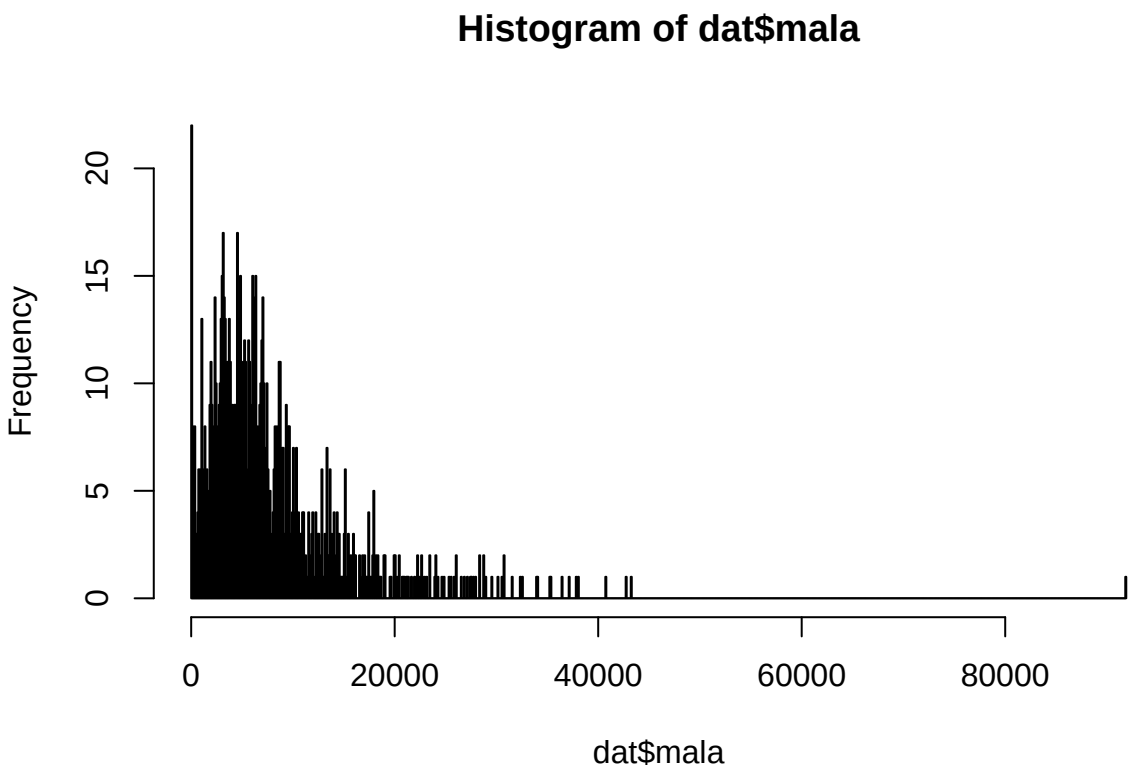

Histogram of dat$mala
Frequency
0
5
10
15
20
0
20000
40000
60000
80000
dat$mala

fig 3

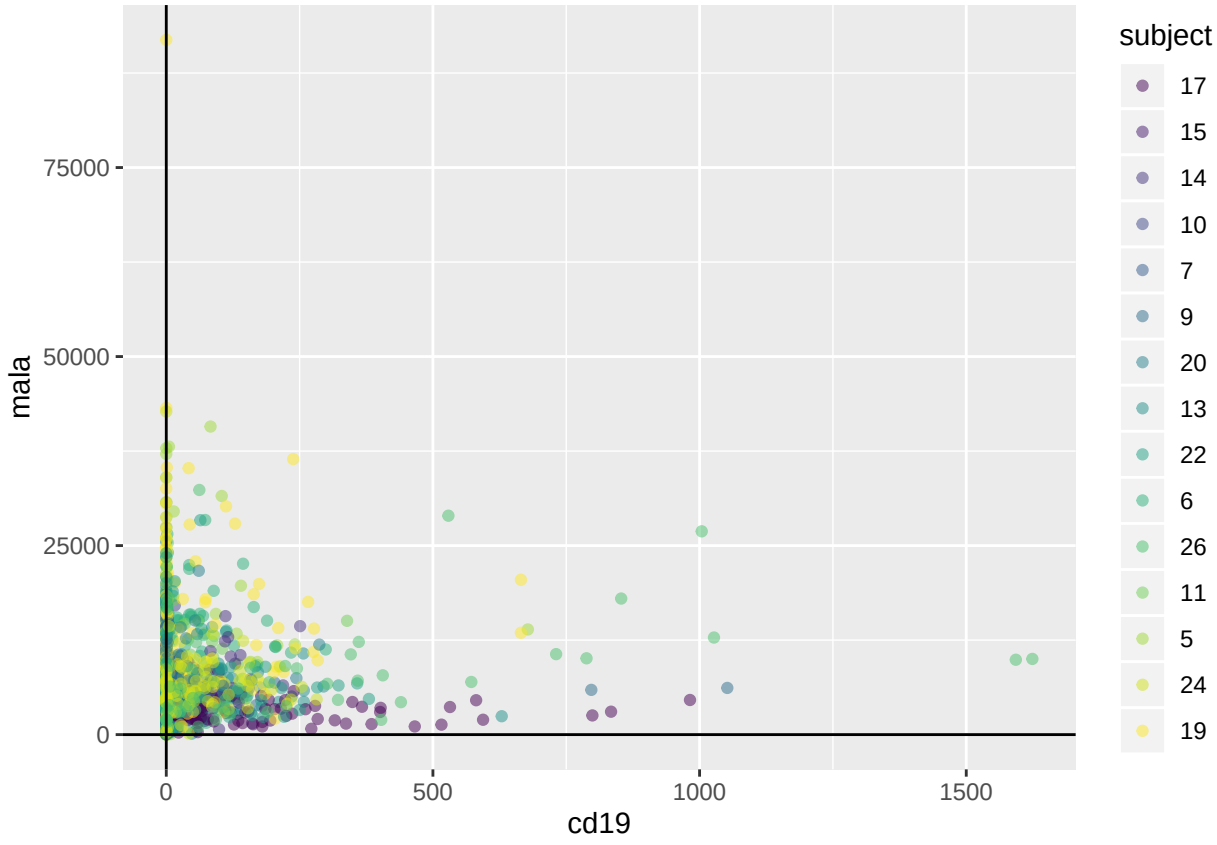

mala
75000
50000
25000
0
0
500
1000
1500
cd19
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19

fig 4

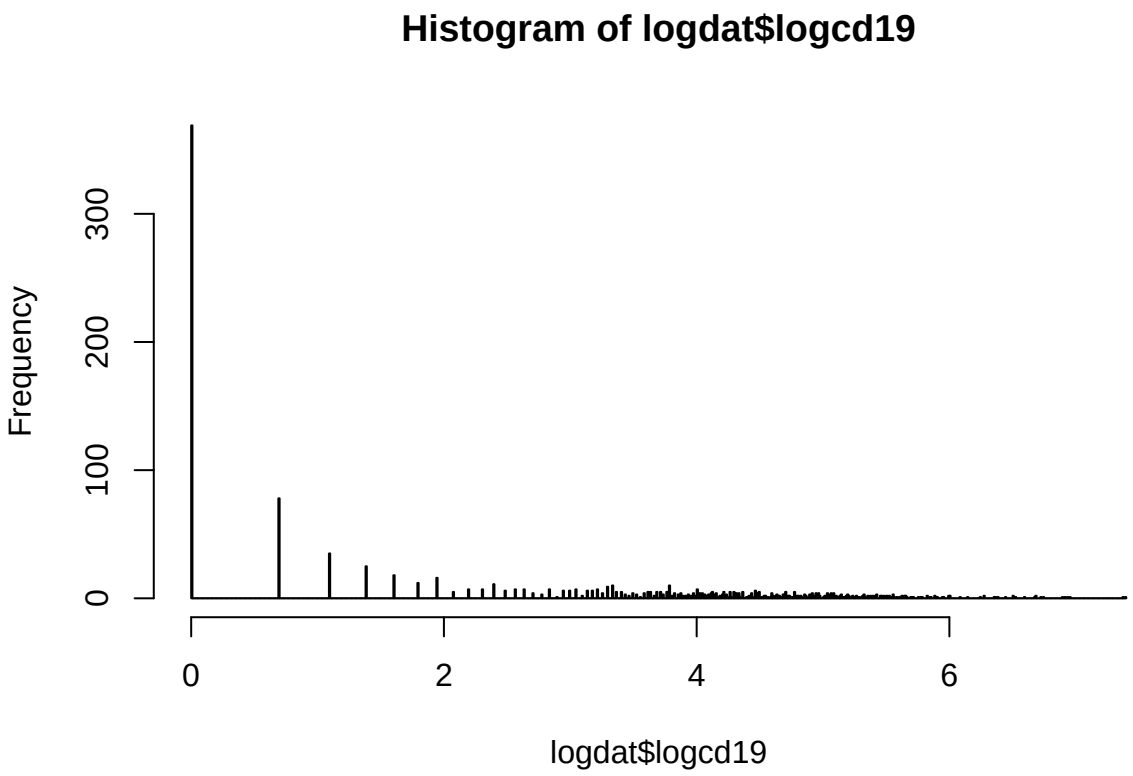

Histogram of logdat$logcd19
Frequency
0
100
200
300
0
2
4
6
logdat$logcd19

fig 5

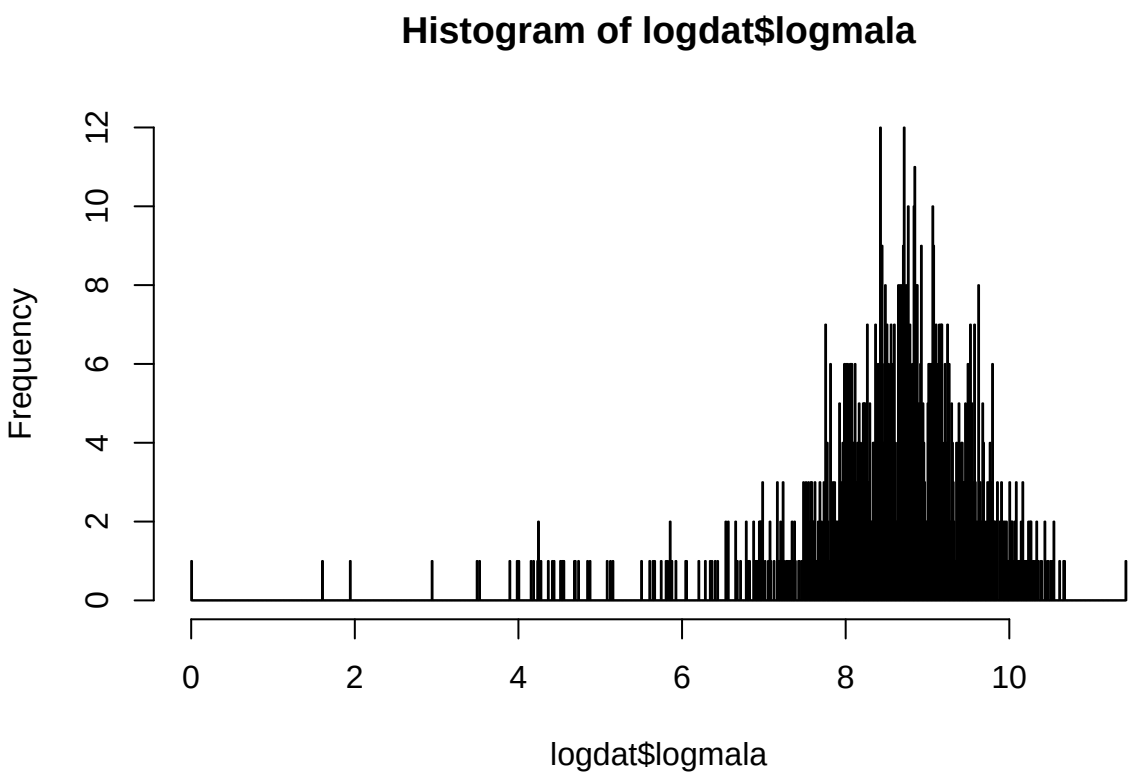
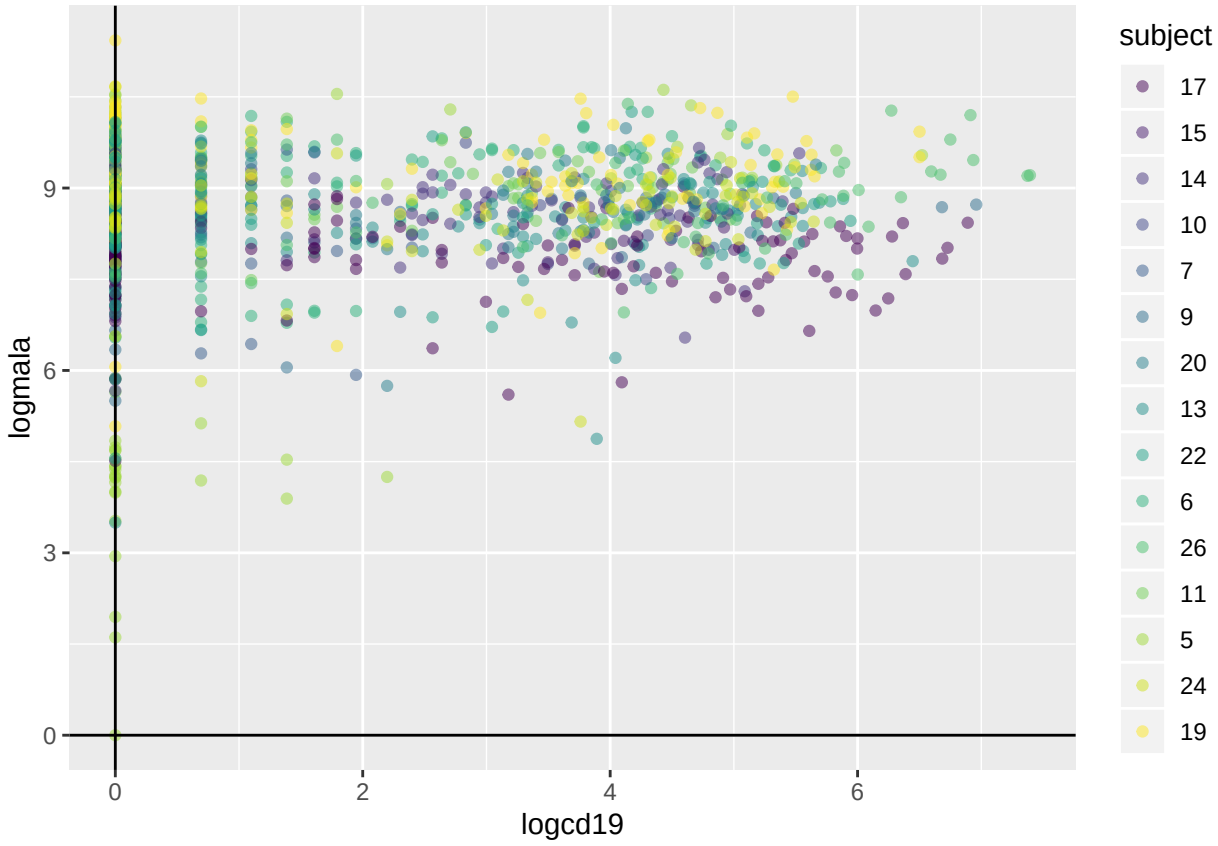
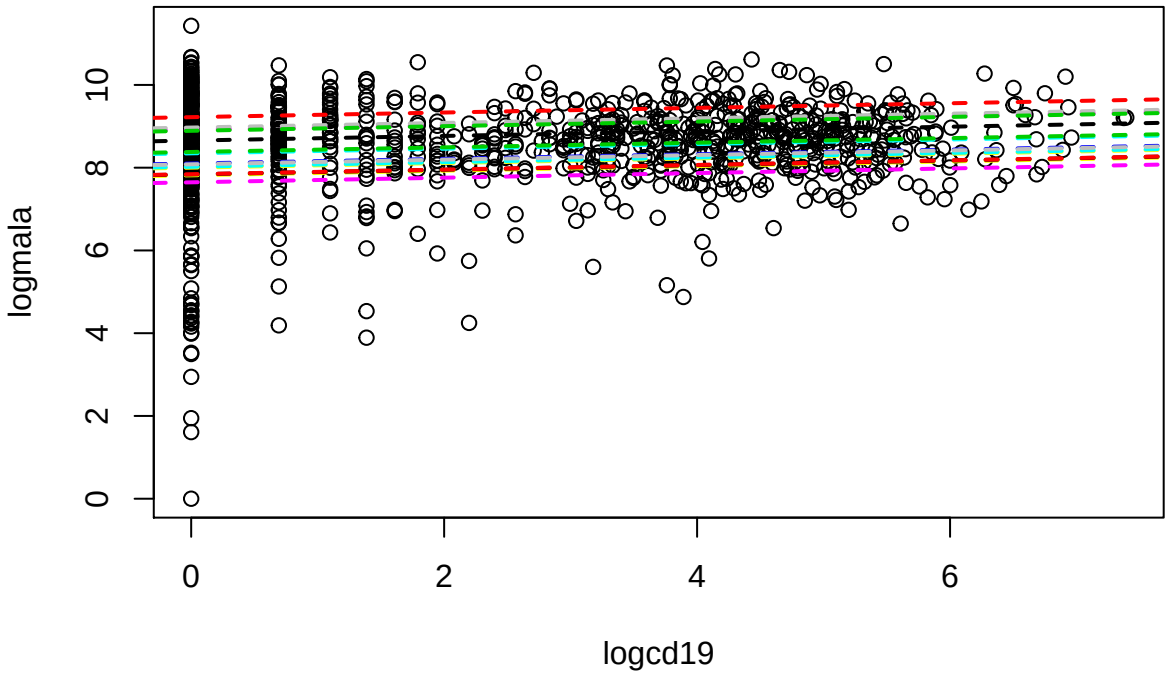
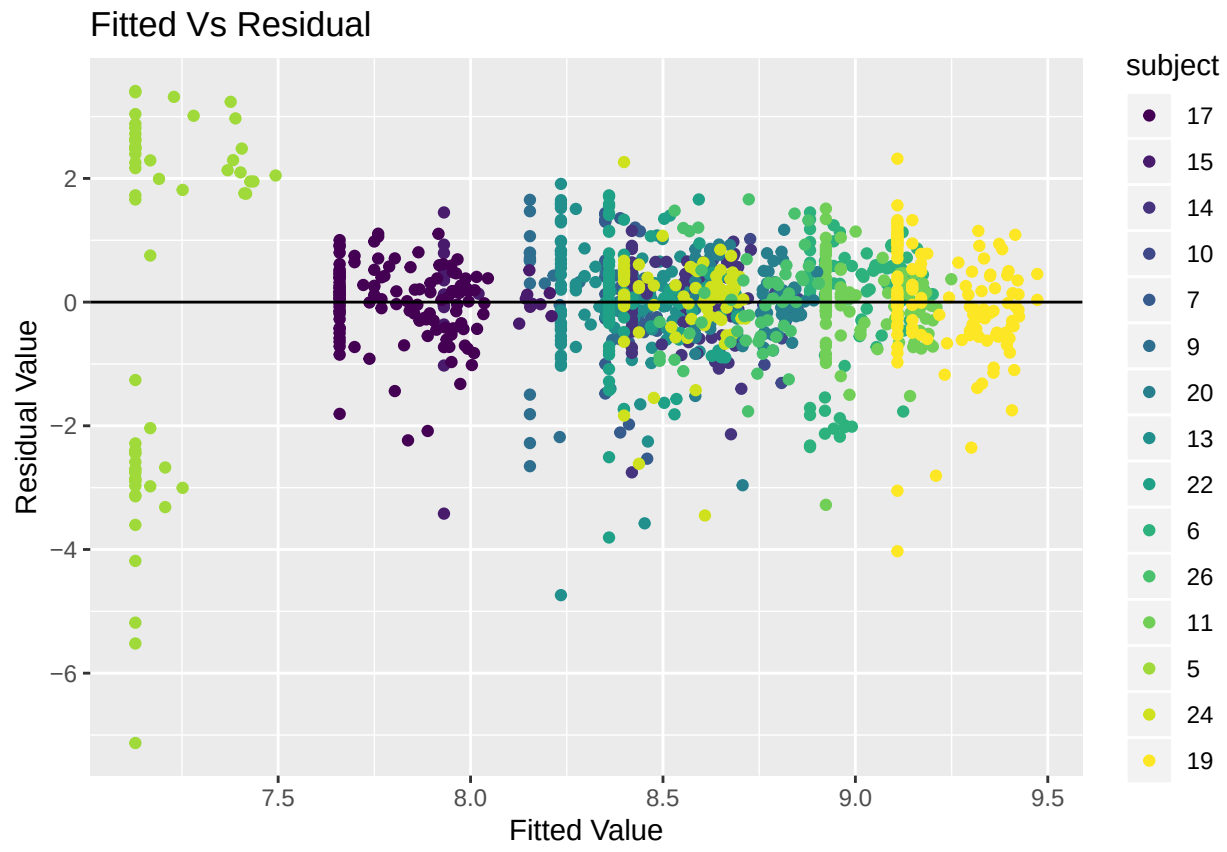

Histogram of logdat$logmala
Frequency
0
2
4
6
8
10
12
0
2
4
6
8
10
logdat$logmala

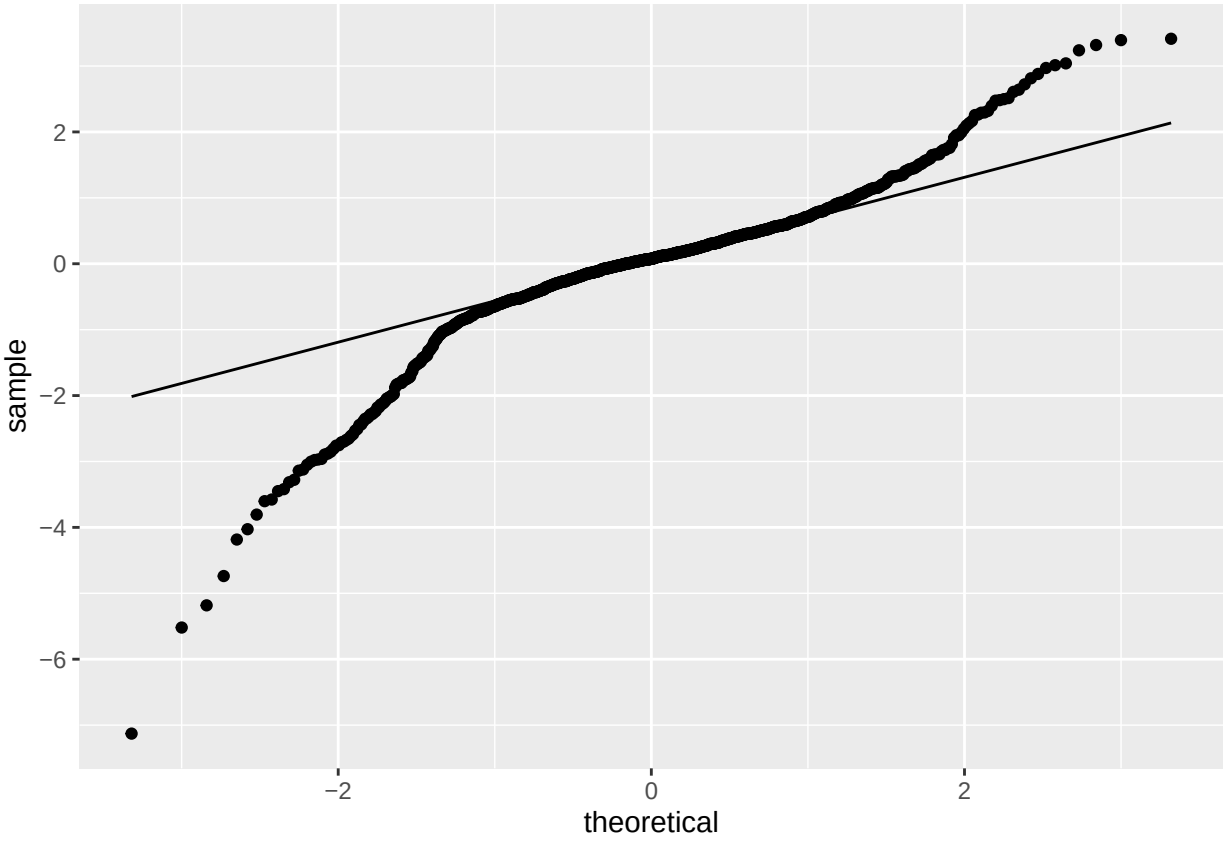
fig 6



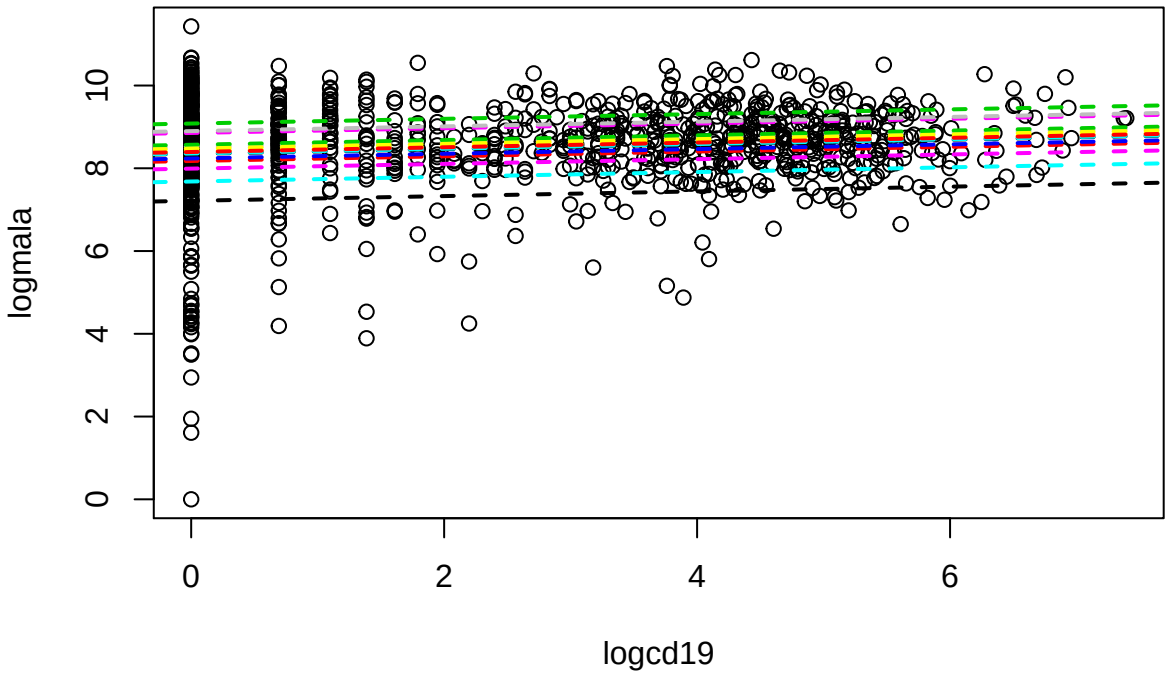
LMwFE Plots

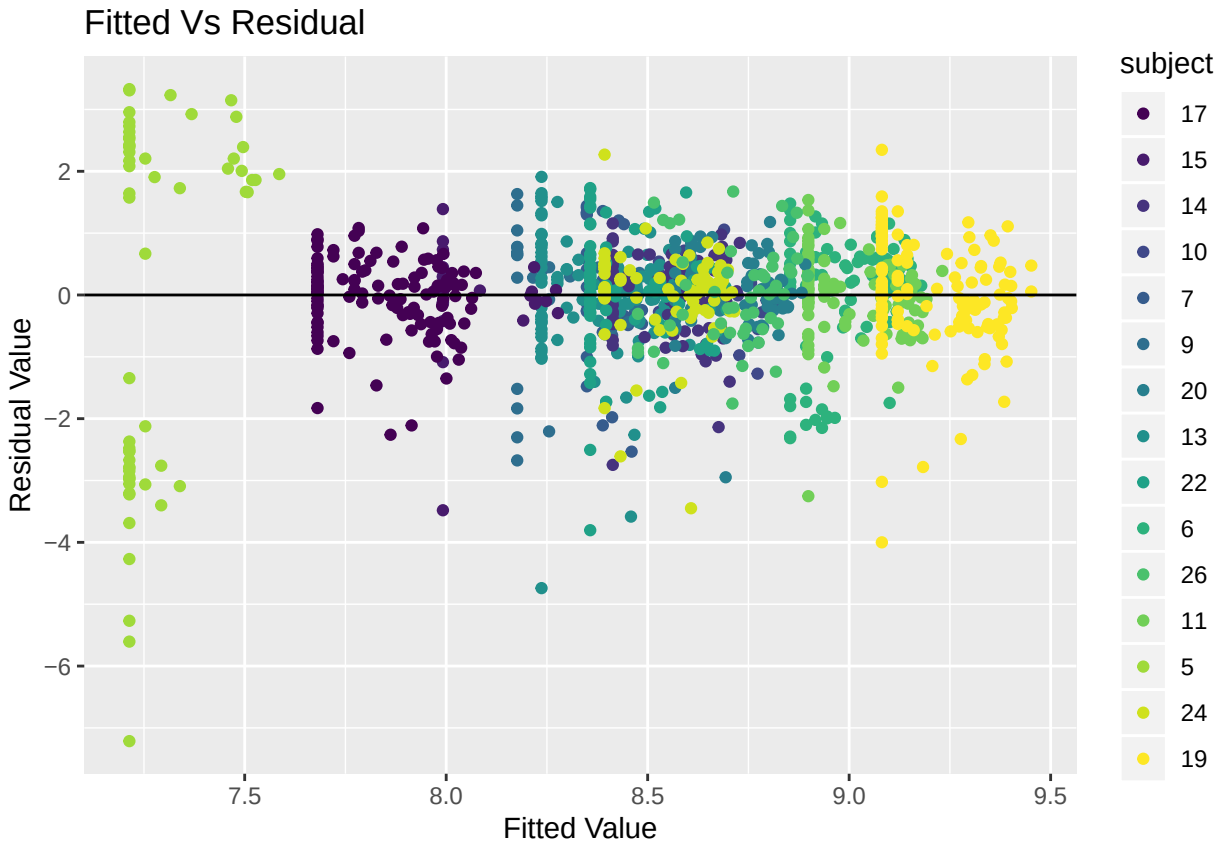


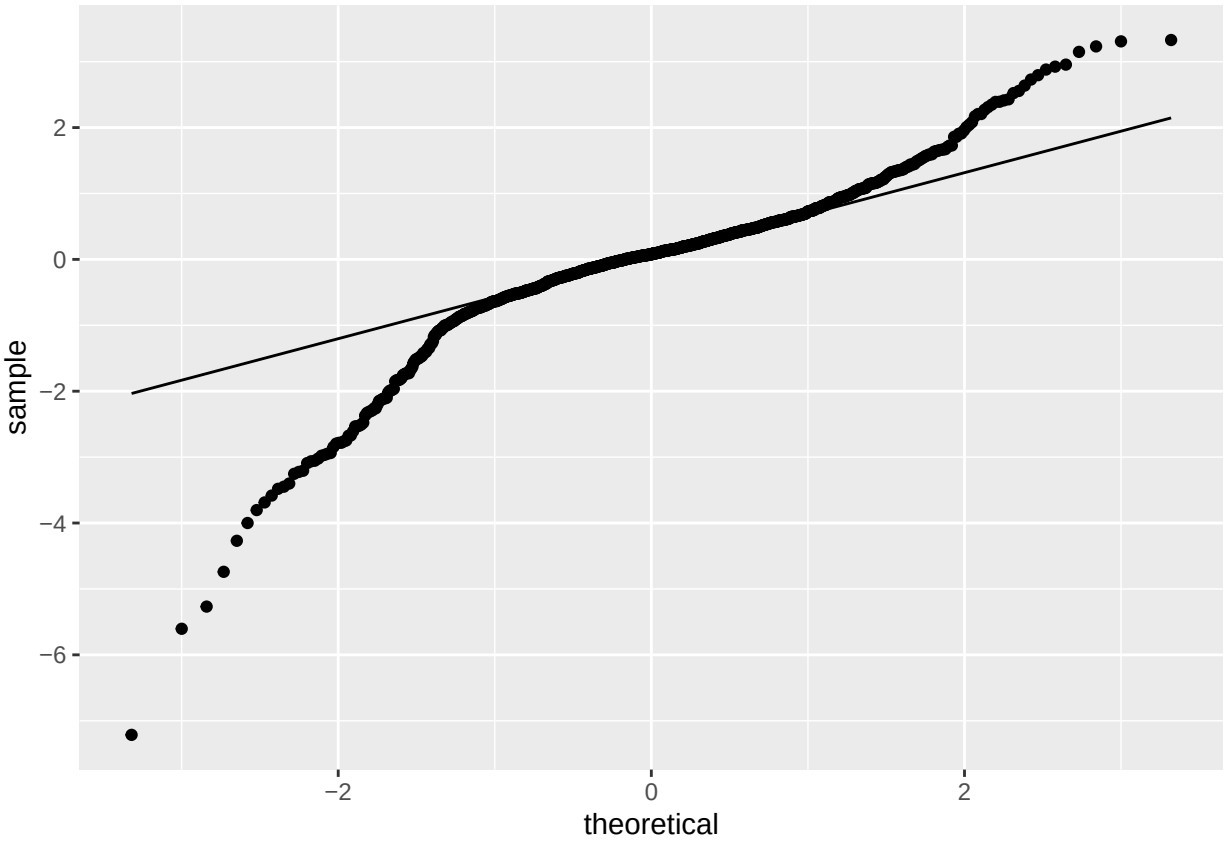

Fitted Vs Residual
Residual Value
Fitted Value
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19


sample
theoretical

LMMwRE Plots

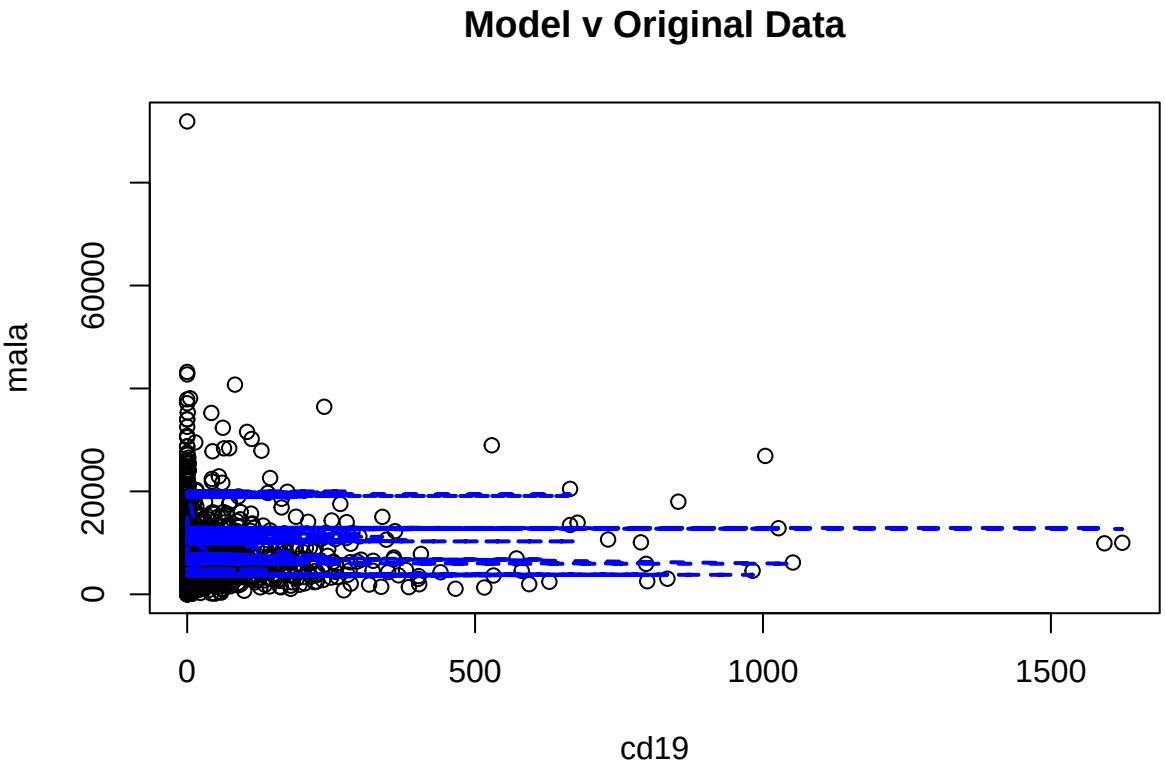


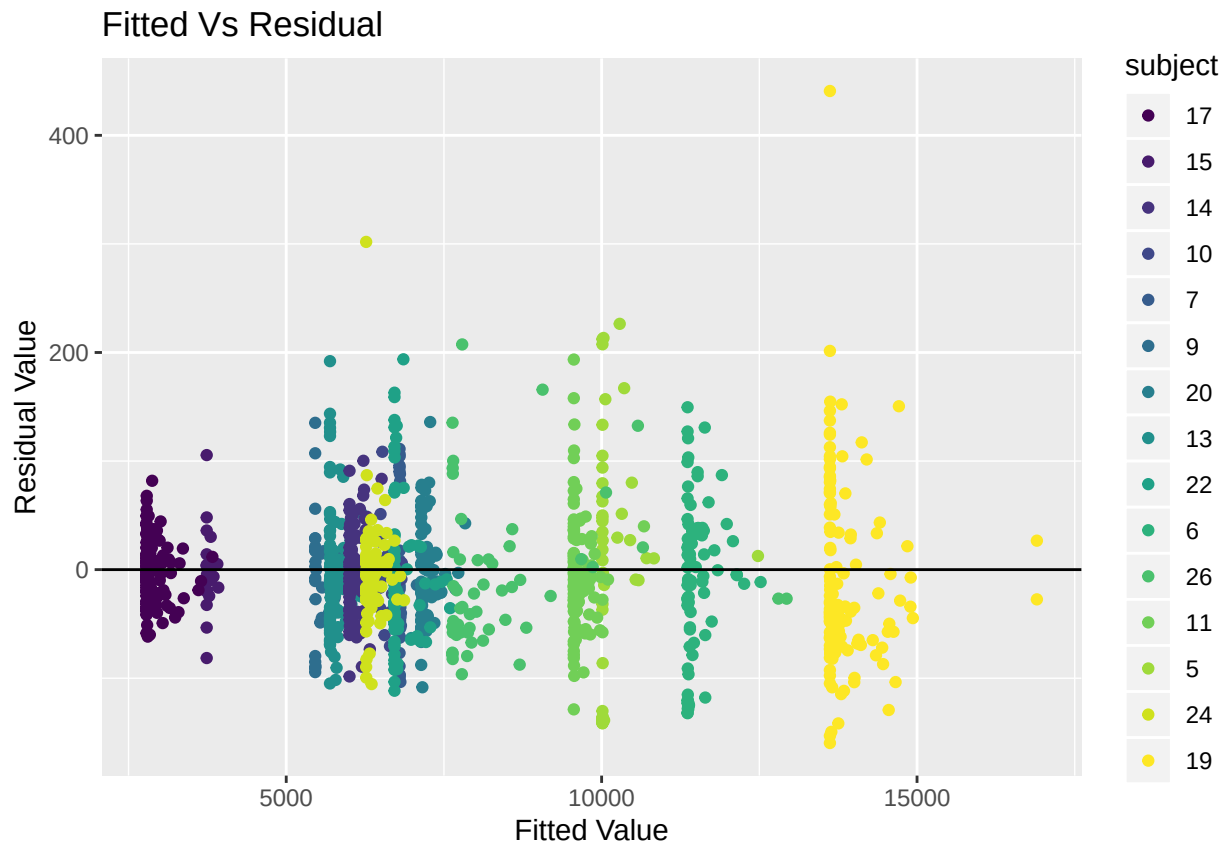


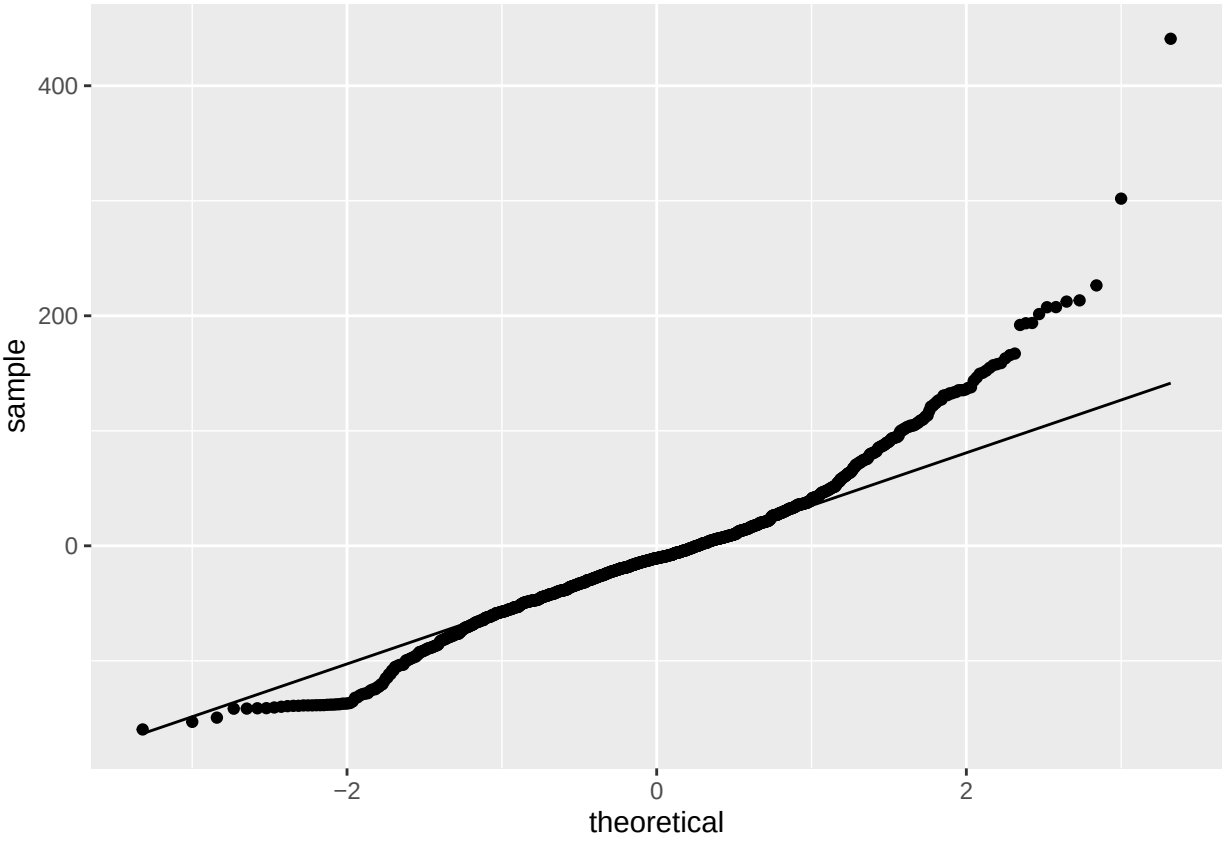


POI Plots

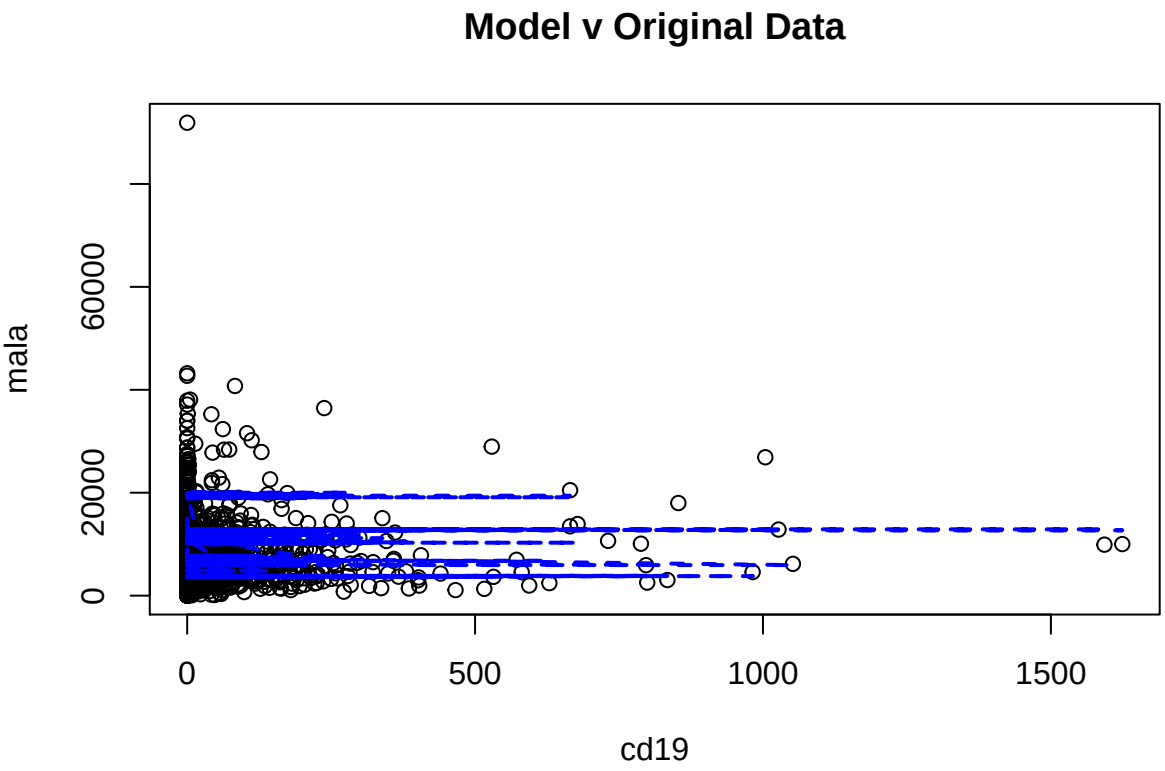
```
## [1]    0 1624
```

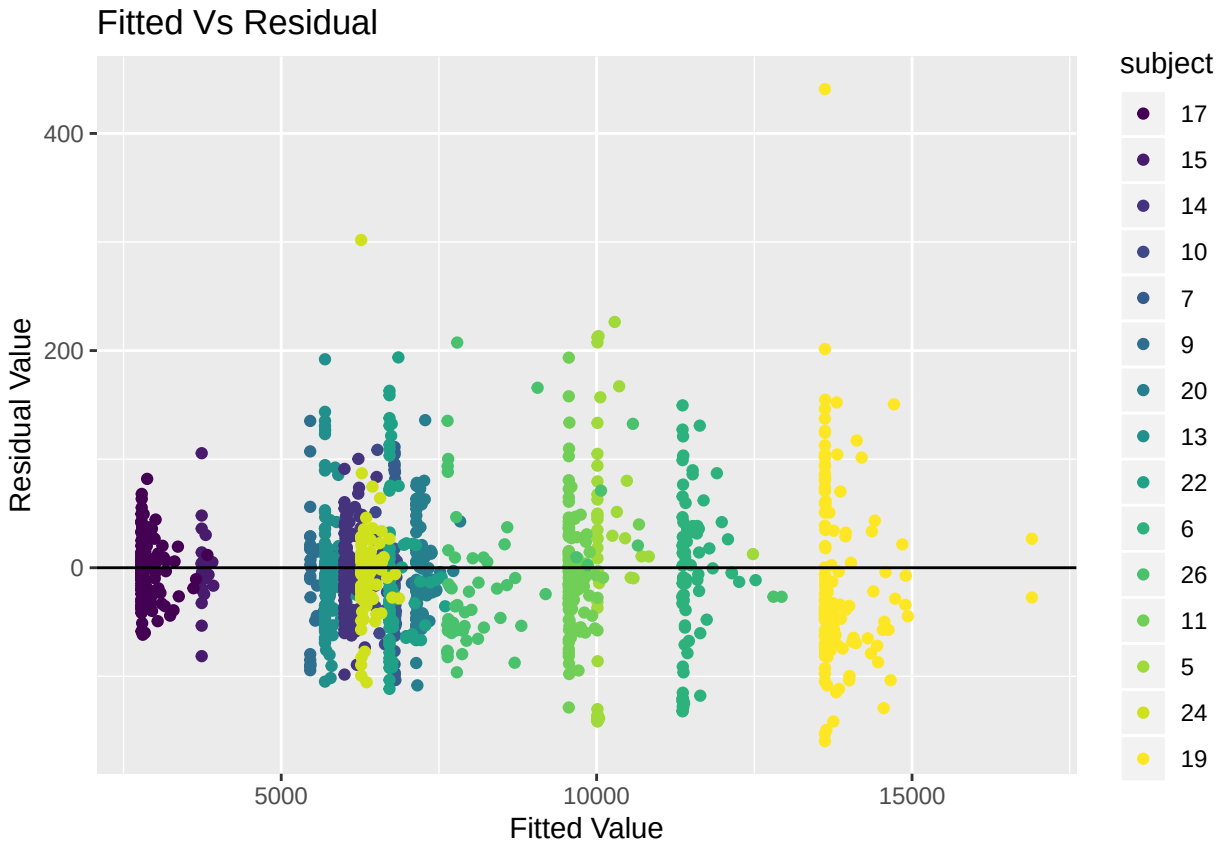


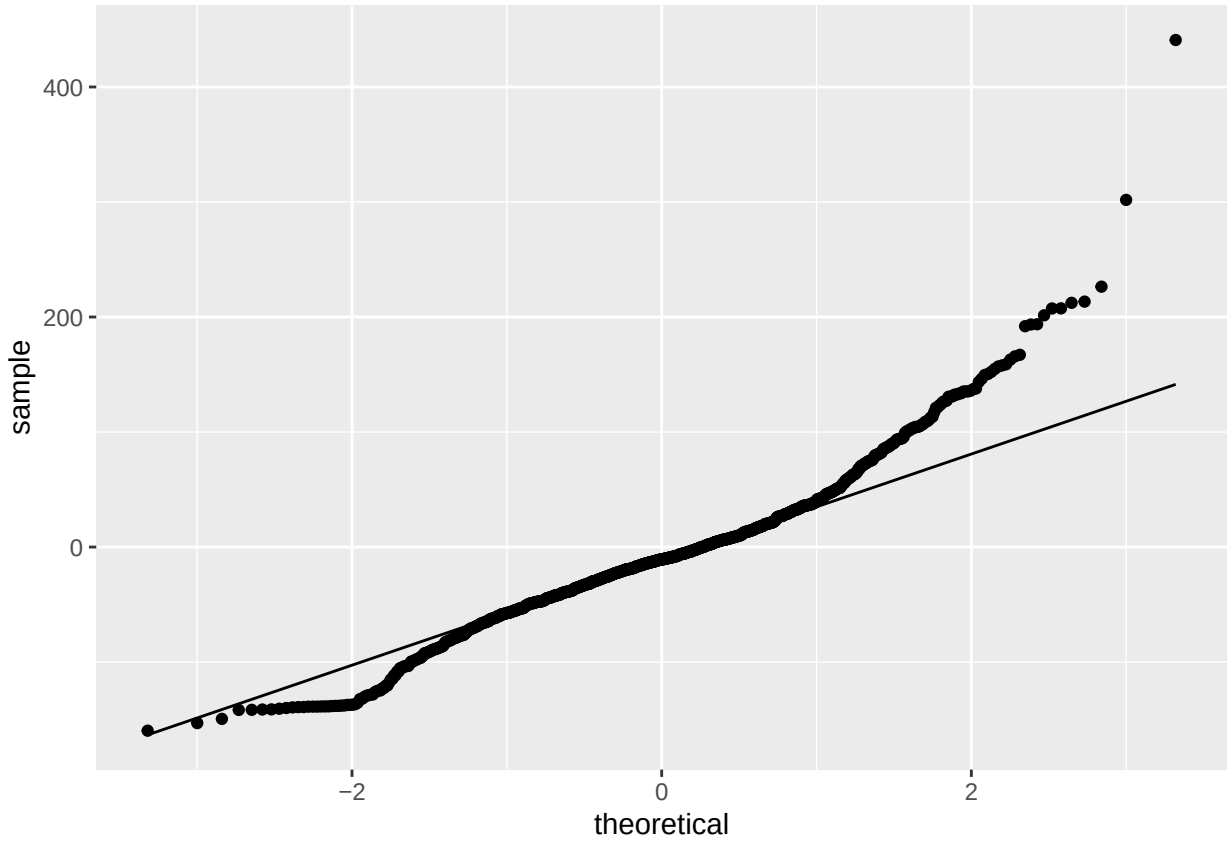

Fitted Vs Residual
Residual Value
400
200
0
5000
10000
15000
Fitted Value
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19


sample
400
200
0
−2
0
2
theoretical

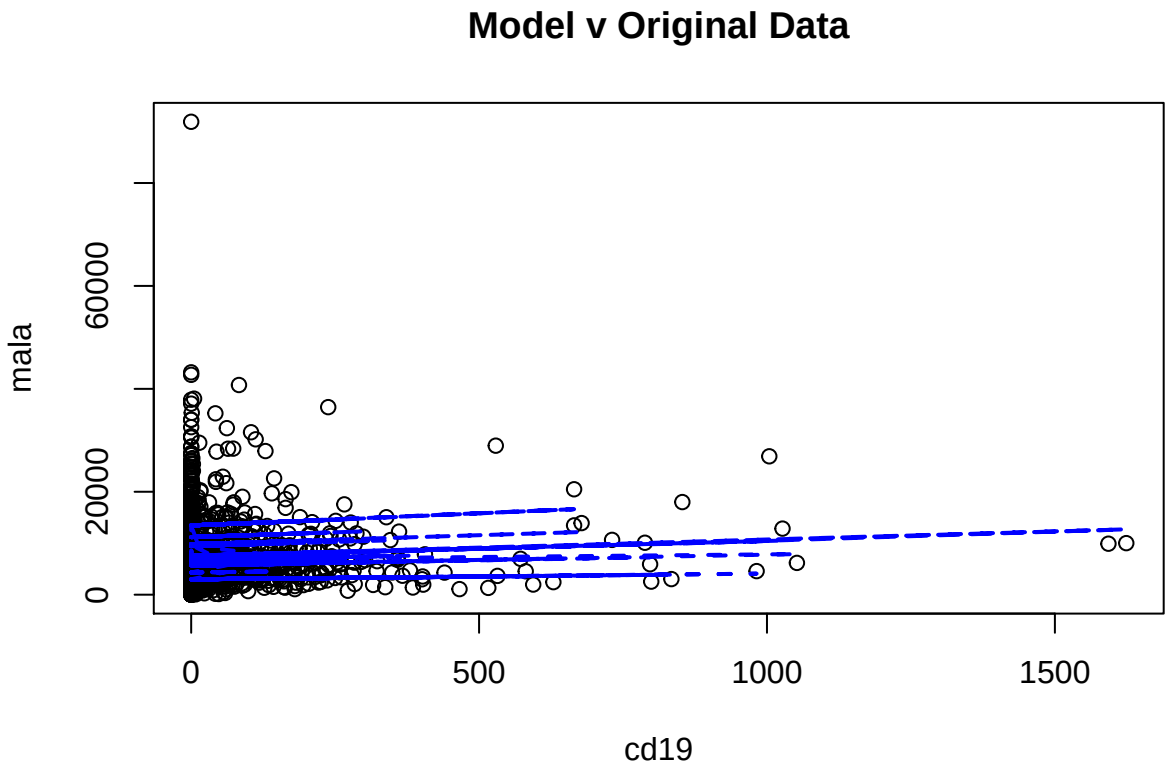
POIql Plots

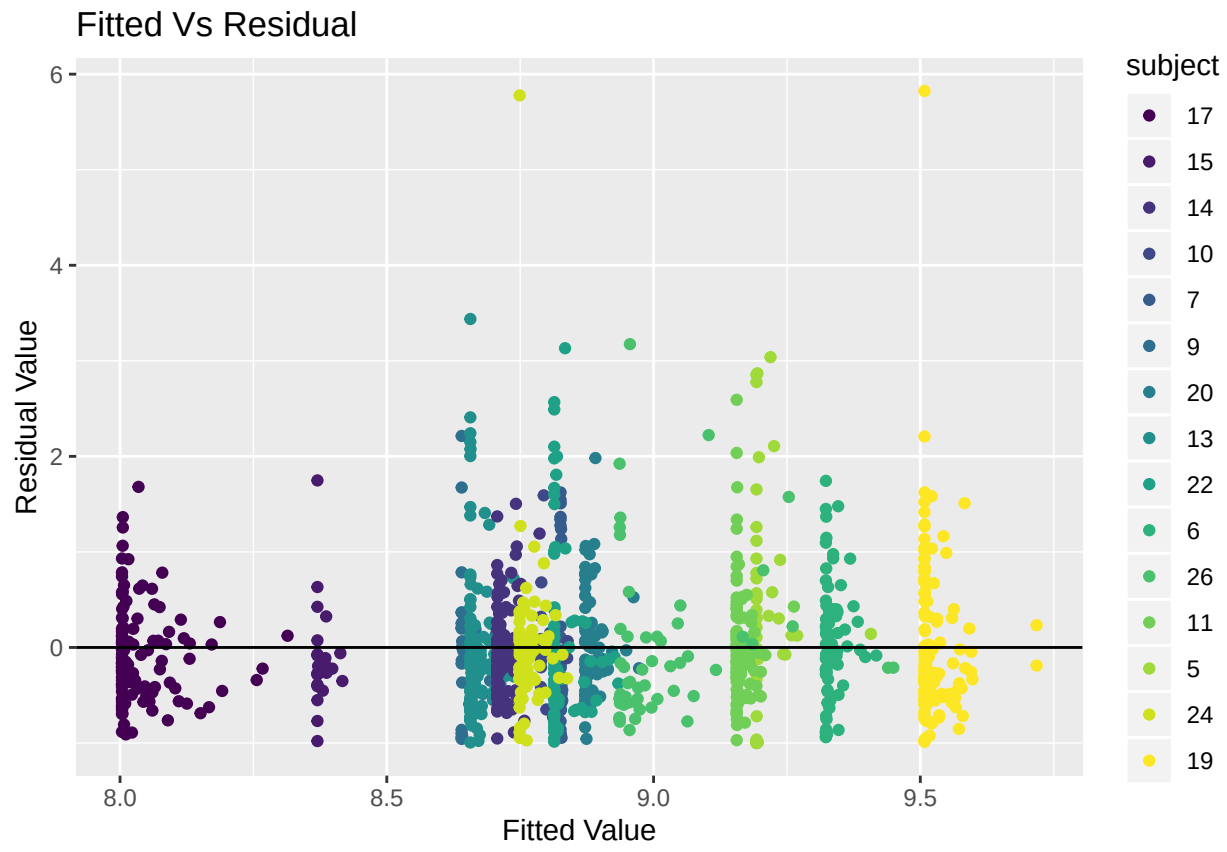

Model v Original Data
mala
0
20000
60000
0
500
1000
1500
cd19

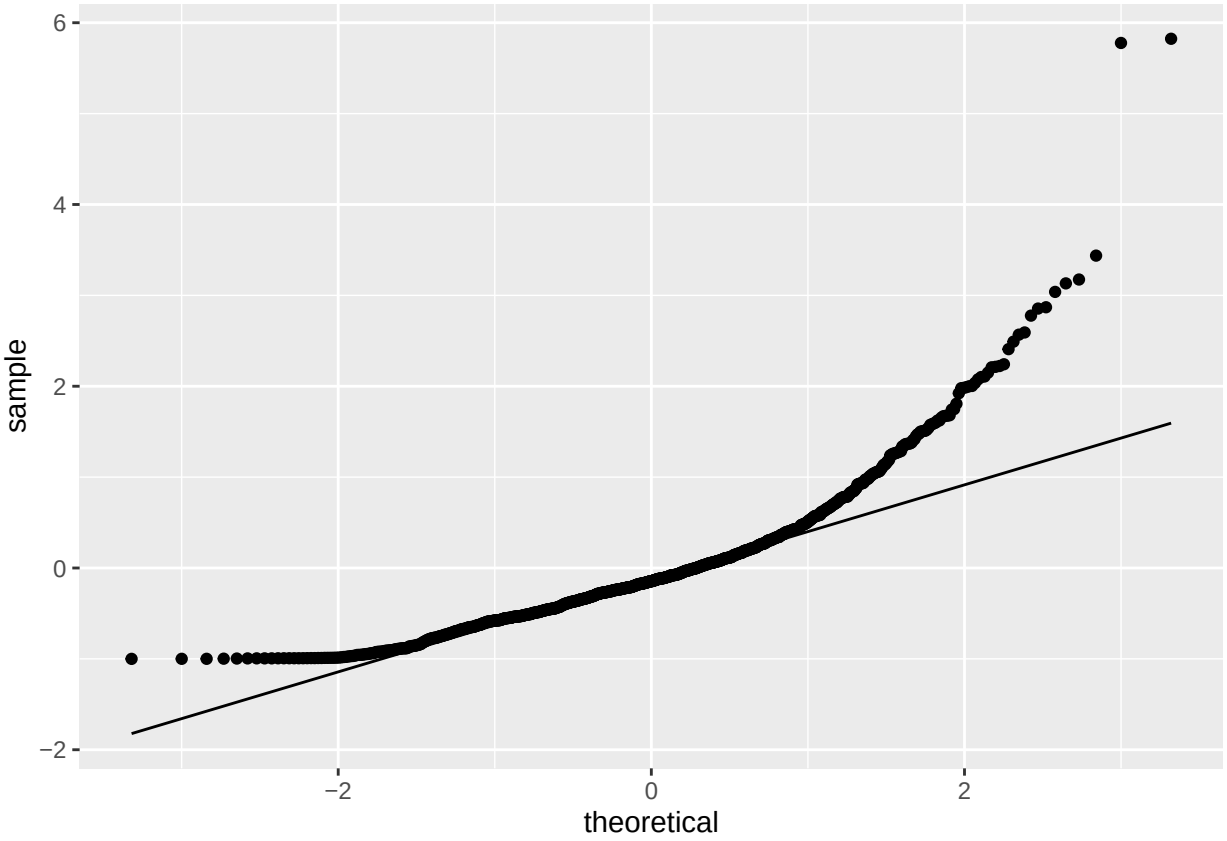

Fitted Vs Residual
Residual Value
400
200
0
5000
10000
15000
Fitted Value
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19



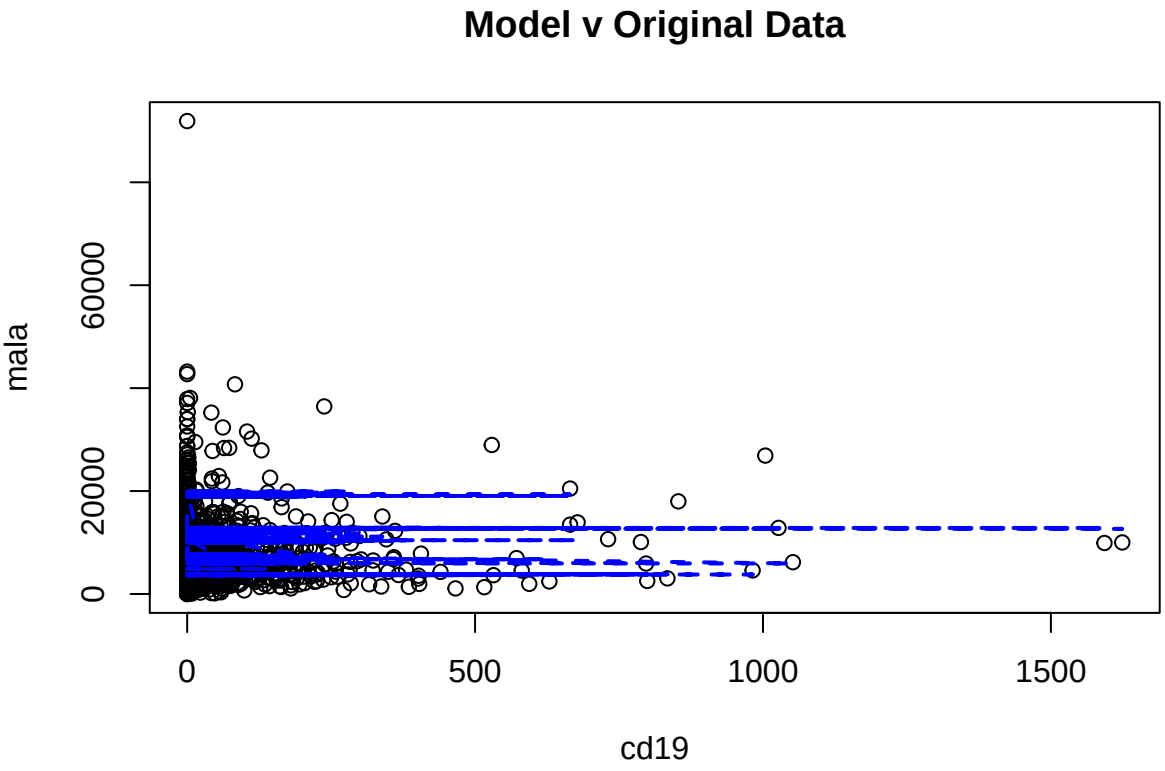
POIlmm Plots

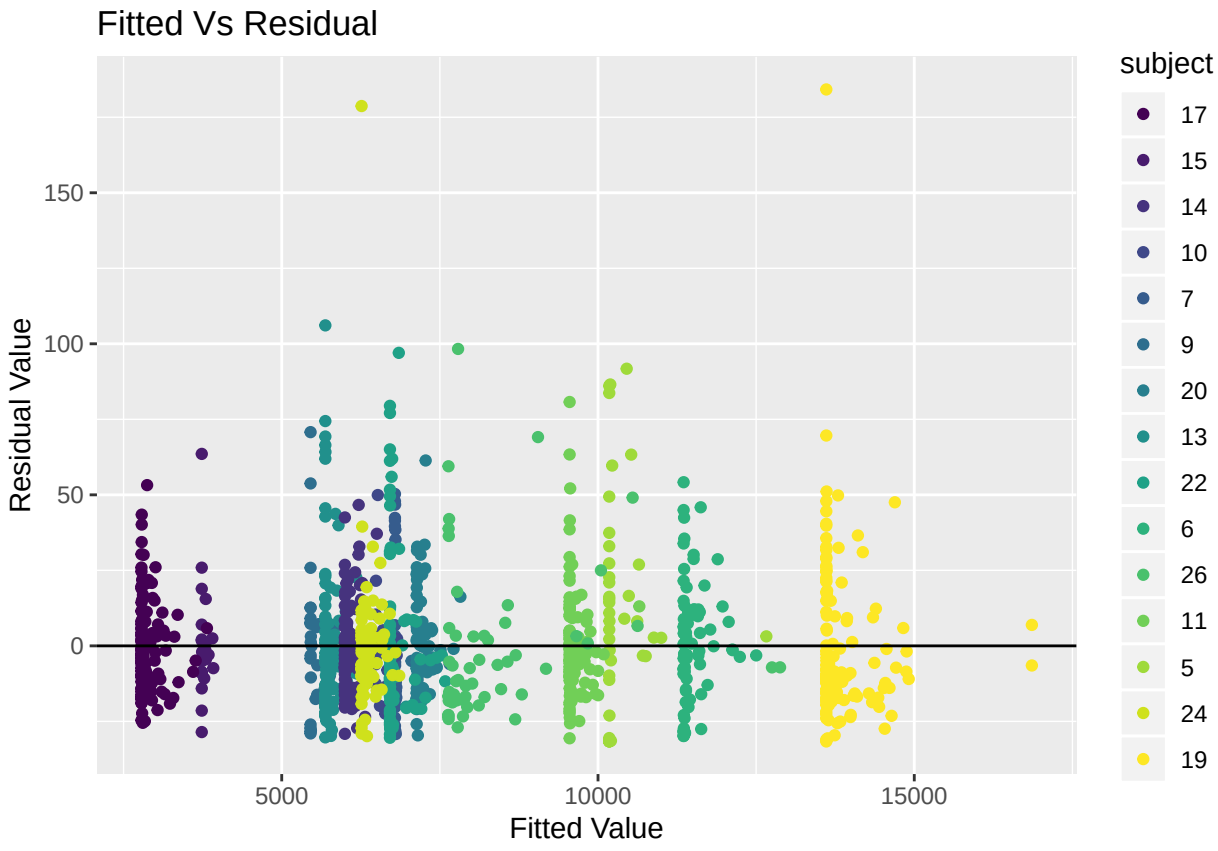


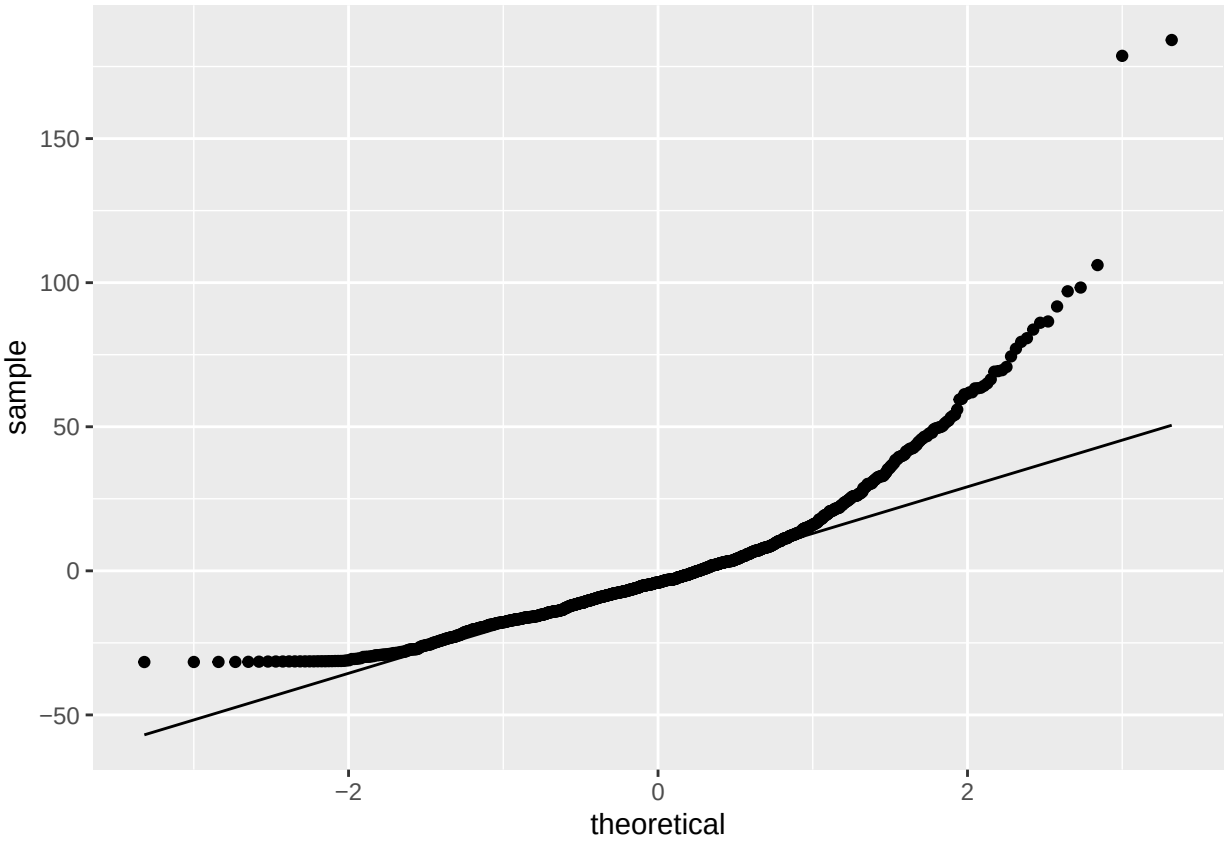

Fitted Vs Residual
Residual Value
Fitted Value
6
4
2
0
8.0
8.5
9.0
9.5
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19


sample
theoretical
6
4
2
0
−2
−2
0
2

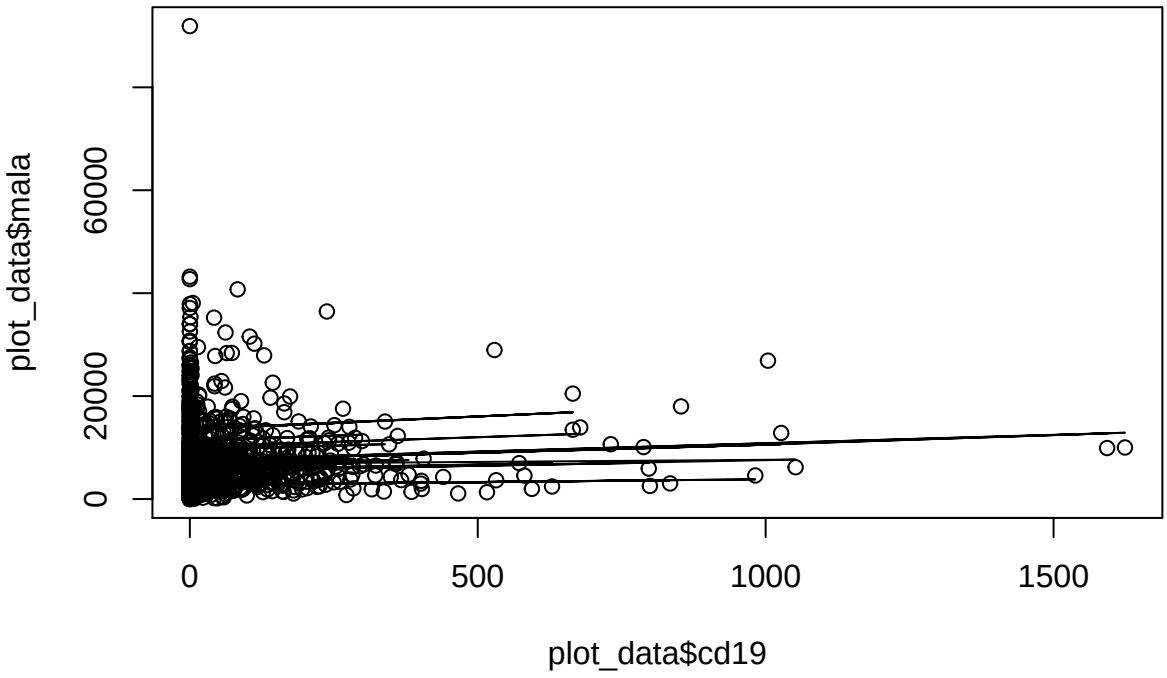
ZIPfe Plots

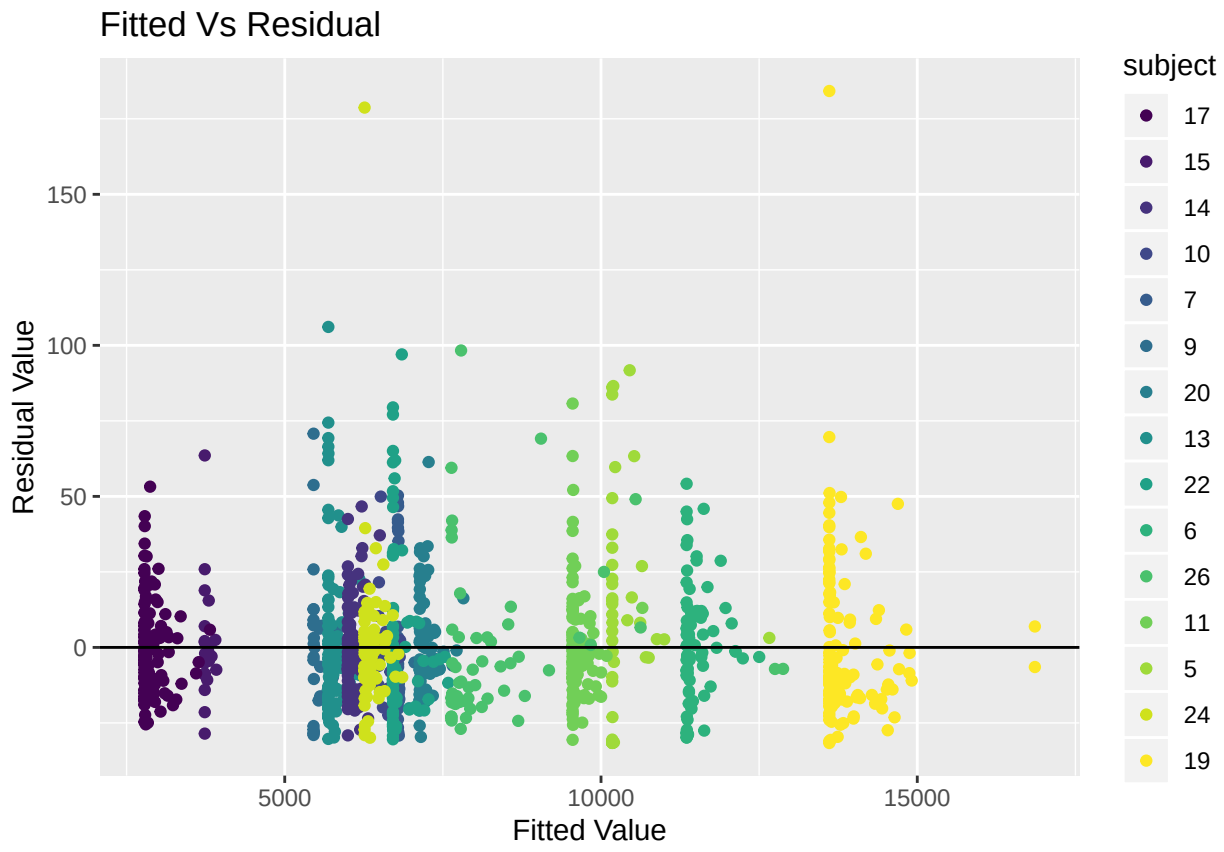

Model v Original Data
mala
0
20000
60000
0
500
1000
1500
cd19

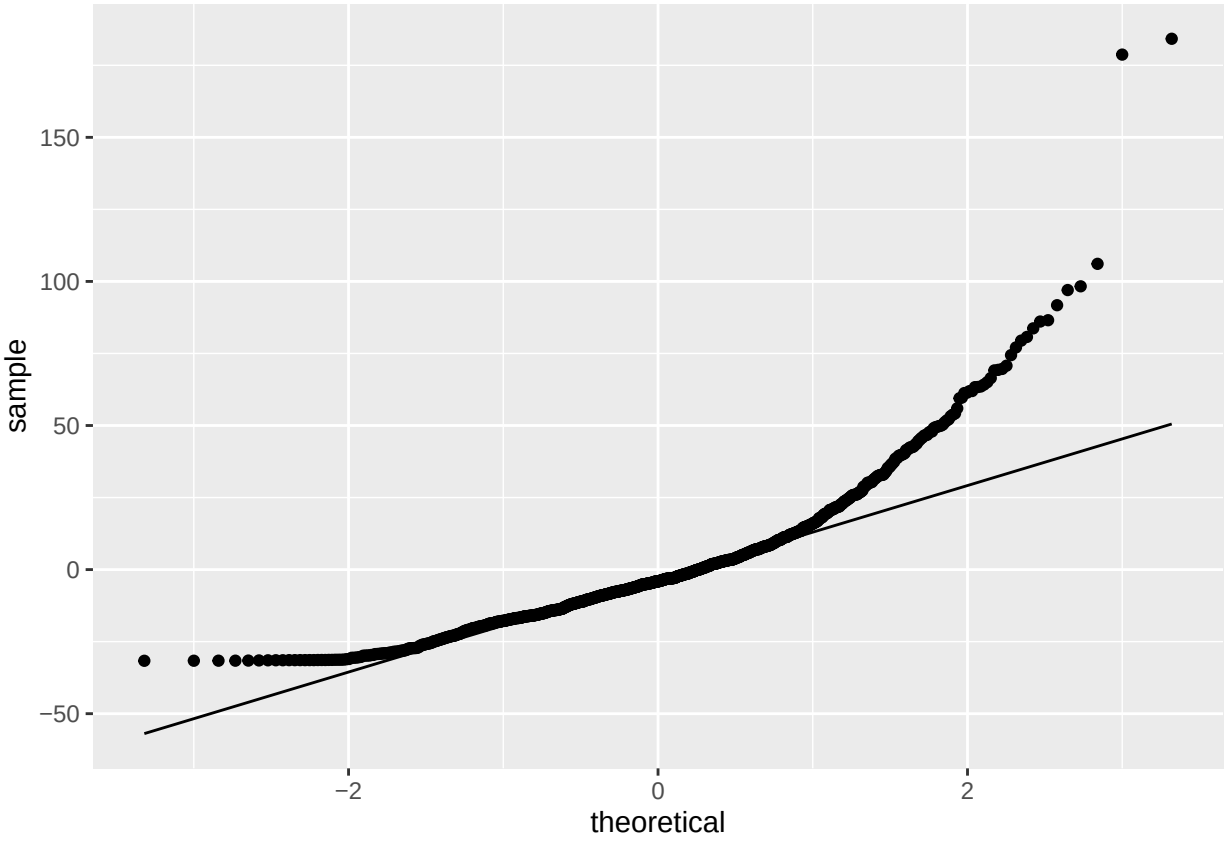

Fitted Vs Residual
Residual Value
150
100
50
0
5000
10000
15000
Fitted Value
subject
17
15
14
10
7
9
20
13
22
6
26
11
5
24
19


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

ZIPre Plots


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

```
####    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)
```

```r
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)+
```

```
  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:


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