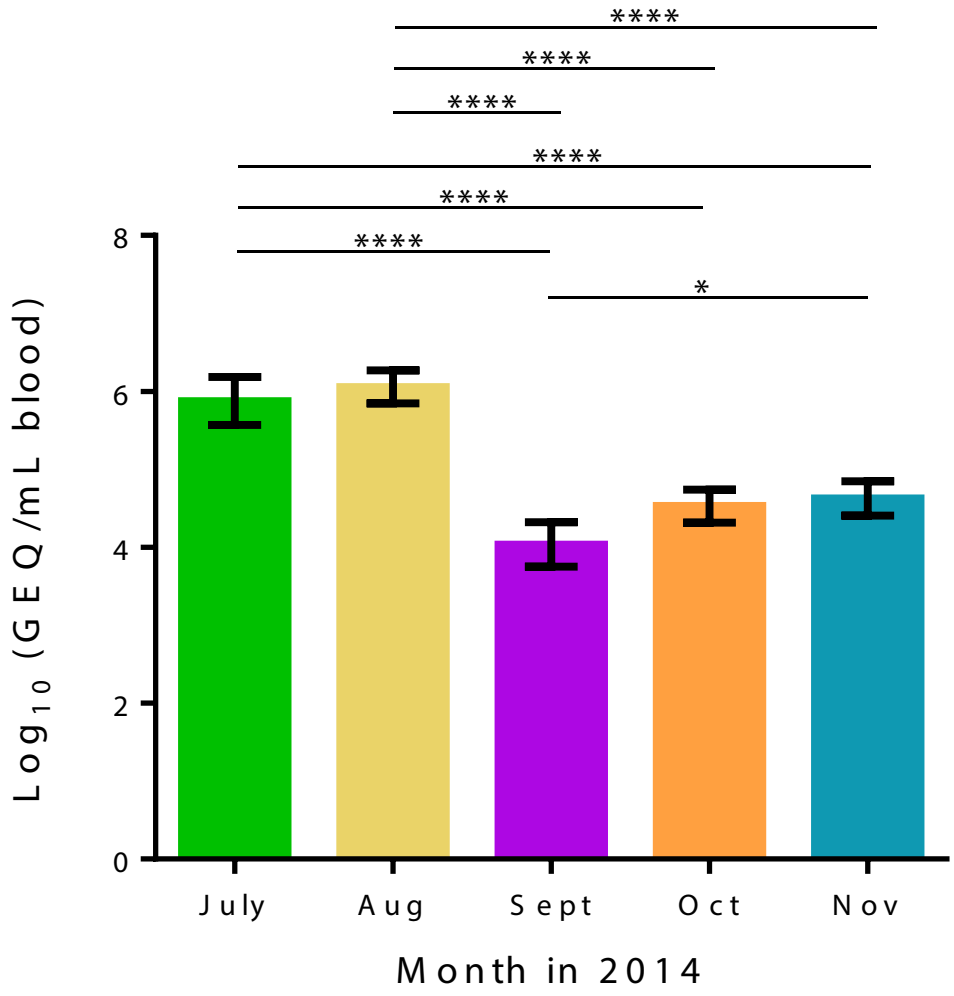
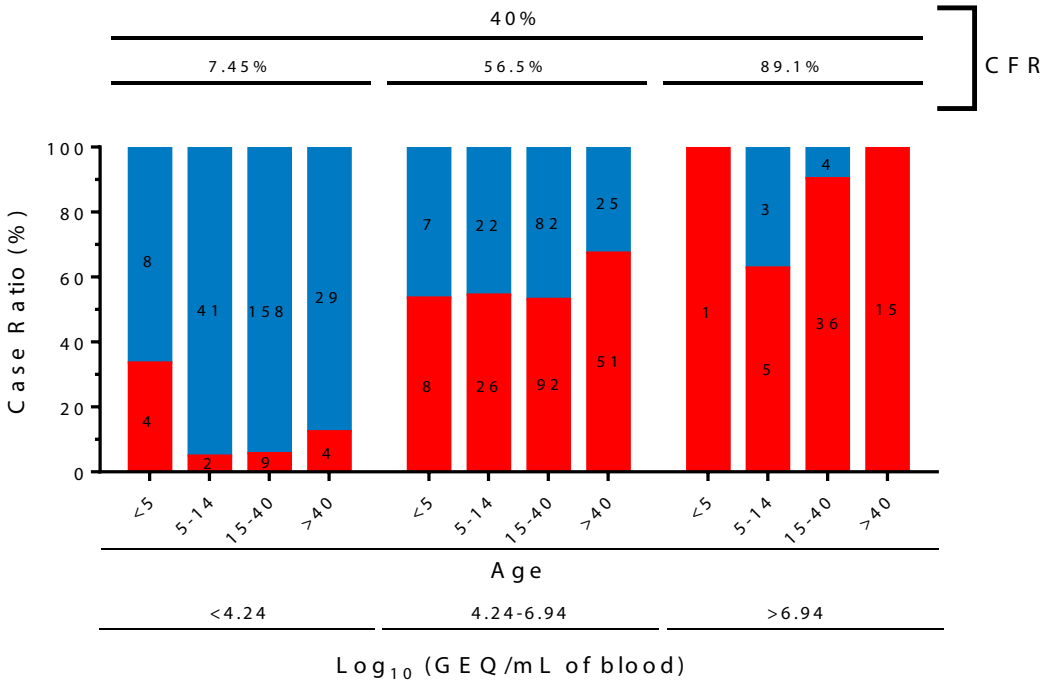


FIGURE S1: Monthly average viremia . (n=632) The mean viremia for each month of the outbreak in Kailahun. Error bars represent the 95% confidence interval for each mean. Ordinary one-way ANOVAs with Tukey's multiple comparisons test****: $p<0.0001$, *: $0.01<p<0.05$



7

8 FIGURE S2: *Correlation between viral load and outcome for the entire data collection*

9 *period.* (n=632) The case fatality rate (CFR, %) and the case survival rate (CSR, %) are

10 stacked, divided by viral clusters of high (>6.94), intermediate (4.24-6.94) and low

11 (<4.24) viremia and sub-divided by age groups (Below 5, between 5 and 14, between 15

12 and 40 and above 40 years old) for the whole outbreak in Kailahun, Sierra Leone, from

13 July to November. Survivors are shown in blue and non-survivors are shown in red.

14

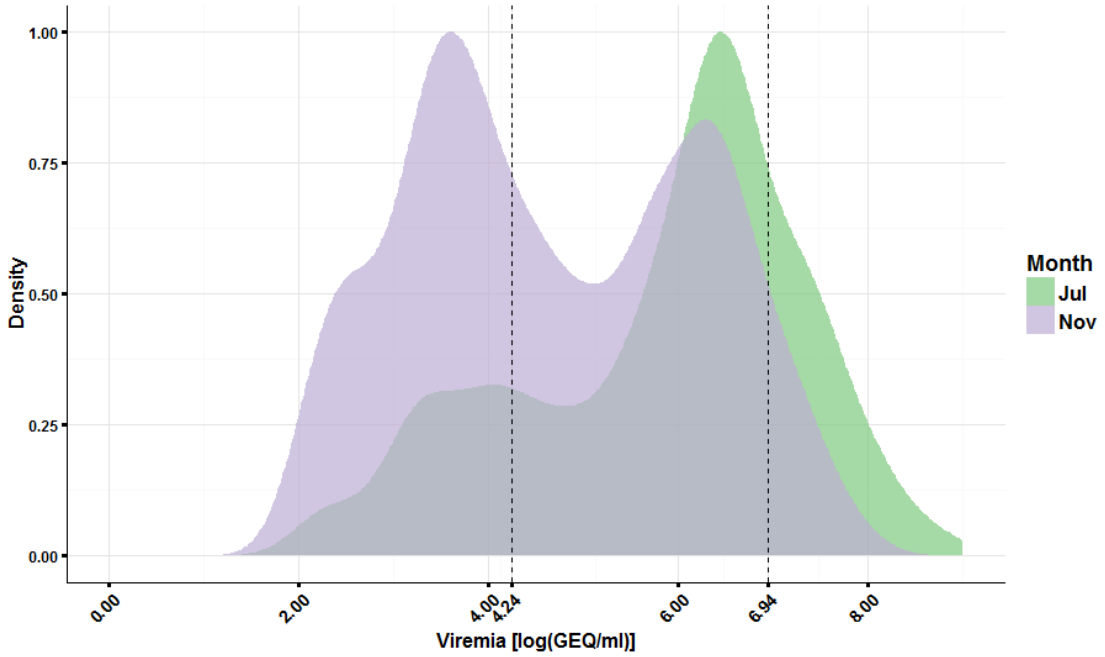


FIGURE S3: *Proportion of cases for the months of July and November*. The proportion of cases for July (Green) and November (Purple) is shown based on viremia. Dashed lines represent the viral clusters defined in the text (< 4.24, 4.24-6.94, > 6.94).

CT	Corresponding Viral load	
	GEQ/mL	Log10(GEQ/mL)
15	1,95E+08	8,29
18	3,02E+07	7,48
20	8,71E+06	6,94
22	2,51E+06	6,40
25	3,89E+05	5,59
30	1,74E+04	4,24
35	7,76E+02	2,89

TABLE S1: *Conversion between CT values obtained by RT-qPCR and the corresponding viral loads*

Month	Log RNA copies/mL of blood July 1st	Log RNA copies/mL of blood End of the month	Log difference [95% CI]
July	5,95	5,72	-0,22 [-1,25; 0,81]
August	5,77	6,08	0,31 [-0,48; 1,10]
September	6,50	4,13	-2,37 [-2,90; -1,84]
October	6,15	4,06	-2,09 [-2,56; -1,61]
November	5,80	4,20	-1,60 [-2,04; -1,17]

TABLE S2: *Log-unit difference between the end of a given month and July 1st in the linear regression*

Month	Threshold value Log_{10} (GEQ/mL of blood)	Sensitivity [95% CI]	Specificity [95% CI]	Positive Predictive Value (PPV)	Negative Predictive Value (NPV)	Area Under the Curve (AUC) [95% CI]
July	5,860	0,694 [0,554; 0,805]	0,864 [0,751; 0,932]	0,810	0,773	0,846 [0,779; 0,914]
August	5,941	0,487 [0,339; 0,638]	0,676 [0,560; 0,773]	0,452	0,706	0,553 [0,438; 0,668]
September	4,793	0,848 [0,765; 0,905]	0,828 [0,708; 0,905]	0,899	0,750	0,867 [0,809; 0,925]
October	4,942	0,840 [0,763; 0,896]	0,824 [0,714; 0,897]	0,893	0,747	0,865 [0,809; 0,922]
November	4,672	0,752 [0,670; 0,818]	0,910 [0,814; 0,961]	0,942	0,656	0,880 [0,830; 0,930]

TABLE S3: *Parameters of the receiver operating characteristic (ROC) analysis*

Tukey's multiple comparisons test	Mean 1	Mean 2	Mean Diff.	95% CI of diff.	Significant?	Summary
July vs. Aug	0,5269	0,6739	-0,147	[-0,3373; 0,04323]	No	ns
July vs. Sept	0,5269	0,3175	0,209	[0,03253; 0,3863]	Yes	*
July vs. Oct	0,5269	0,327	0,200	[0,03093; 0,3688]	Yes	*
July vs. Nov	0,5269	0,3091	0,218	[0,05002; 0,3856]	Yes	**
Aug vs. Sept	0,6739	0,3175	0,357	[0,1790; 0,5339]	Yes	****
Aug vs. Oct	0,6739	0,327	0,347	[0,1774; 0,5164]	Yes	****
Aug vs. Nov	0,6739	0,3091	0,365	[0,1965; 0,5332]	Yes	****
Sept vs. Oct	0,3175	0,327	-0,010	[-0,1639; 0,1447]	No	ns
Sept vs. Nov	0,3175	0,3091	0,008	[-0,1447; 0,1615]	No	ns
Oct vs. Nov	0,327	0,3091	0,018	[-0,1258; 0,1617]	No	ns

TABLE S4: *Parameters of the Tukey's multiple comparisons test ns: not significant, *: 0.01<p<0.05, **:0.001<p<0.01, ****: p<0.0001*

Titer	July	September	October	November	Total
0	6	15	13	15	49
100	1	11	2	6	20
400	0	3	0	4	7
1600	0	4	7	2	13
6400	0	1	3	1	5
Total	7	34	25	28	94

TABLE S5: *Number of patients tested for IgG levels by ELISA and their titers for the given months*

Annex I

EBOV-Makona change in viremia

Jonathan Audet, Marc-Antoine de La Vega

Friday, October 30, 2015

Introduction

Our goal is to evaluate whether the blood viremia changes over two variables: 1) The time since the data collection began (the outbreak itself began before data collection); 2) survivors vs non-survivors, we expect that survivors will generally have lower viremia, as uncontrolled viral replication is usually observed in animals before death.

We will restrict our analysis to the first sample measured after the patients arrived at a treatment center and to cases that are considered "confirmed" for EBOV disease.

Three other variables are available for us to control: age (in years), sex, and time since onset of symptoms. The date of onset of symptoms is estimated by the patients/doctors and may carry significant error.

```
library(ggplot2)
library(reshape2)
library(plyr)
library(dplyr)
library(data.table)
library(GGally)

# multiplot function taken from
# http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

# Multiple plot function ggplot objects can be passed in ..., or to plotlist
# (as a list of ggplot objects) - cols: Number of columns in layout -
# layout: A matrix specifying the layout. If present, 'cols' is ignored.  If
# the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then
# plot 1 will go in the upper left, 2 will go in the upper right, and 3 will
# go all the way across the bottom.
multiplot <- function(..., plotlist = NULL, file, cols = 1, layout = NULL) {
    library(grid)

    # Make a list from the ... arguments and plotlist
    plots <- c(list(...), plotlist)

    numPlots = length(plots)

    # If layout is NULL, then use 'cols' to determine layout
    if (is.null(layout)) {
```

```
        # Make the panel ncol: Number of columns of plots nrow: Number of
rows
        # needed, calculated from # of cols
        layout <- matrix(seq(1, cols * ceiling(numPlots/cols)), ncol = cols,
            nrow = ceiling(numPlots/cols))
    }

    if (numPlots == 1) {
        print(plots[[1]])

    } else {
        # Set up the page
        grid.newpage()
        pushViewport(viewport(layout = grid.layout(nrow(layout),
ncol(layout))))

        # Make each plot, in the correct location
        for (i in 1:numPlots) {
            # Get the i,j matrix positions of the regions that contain this
subplot
            matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

            print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
layout.pos.col = matchidx$col))
        }
    }
}
```

Makona outbreak in Sierra Leone

Load and transform data

Load the clinical dataset and the dataset of previous PCR standards.

```
data <- read.csv("../Data/Marc/AssembledData.csv")
head(data)
```

```
##   Patient_ID   Sampling Lab_number PCR1 PCR1.pos PCR2 PCR2.pos Sex
## 1        460 2014-04-01       1422 33.9        1 34.4        1   F
## 2          8 2014-07-01          1 19.0        1   NA       NA   M
## 3          9 2014-07-01          2 22.0        1   NA       NA   M
## 4          6 2014-07-02          6 26.0        1   NA       NA   F
## 5         10 2014-07-02          5 22.0        1   NA       NA   M
## 6         11 2014-07-02          4 23.0        1   NA       NA   M
##   Age_years  Admission      Onset Outcome_Date Outcome
## 1        40 2014-09-10 2014-09-03   2014-10-16   Cured
## 2        35 2014-06-30 2014-06-27   2014-07-02   Death
## 3         3 2014-06-30 2014-06-28   2014-07-07   Death
## 4        50 2014-06-29 2014-06-20   2014-07-15   Cured
```

```
## 5        50 2014-07-01 2014-06-23   2014-07-03   Death
## 6        23 2014-07-01 2014-06-26   2014-07-03   Death
```

```
pcr.std <- read.csv("../Data/Marc/pcr standards.csv")
head(pcr.std)
```

```
##       RNA       X     X.1     X.2     X.3     X.4     X.5     X.6     X.7
## 1 1.9e+07 11.0356  9.2624 12.2103 12.9507 11.8034 12.0229 11.4621 11.9850
## 2 1.9e+06 14.1855 11.9886 15.4917 16.4511 14.9848 15.4588 15.3741 15.7534
## 3 1.9e+05 17.5014      NA 19.4319 20.0837 18.9865 18.6363 19.0388 19.2609
## 4 1.9e+04 21.4667 18.1868 23.4490      NA 22.8259 22.7607 22.9960 22.8139
## 5 1.9e+03 24.5796 22.1327 26.6000 27.2702 26.1694 26.1786 25.9699 26.0802
## 6 1.9e+02 28.3589 26.8955 30.6951 31.0794 30.1012 30.0749 29.8007 30.6631
##       X.8     X.9
## 1 11.1495  9.6651
## 2 15.1983 14.6601
## 3 19.0694 16.6507
## 4 22.7298 21.5866
## 5 26.0908 25.5301
## 6 30.0601 29.2903
```

```
# Reformat the pcr standards for regression, column 2 is the 'variable name'
# column and is useless here
pcr.std <- melt(pcr.std, id.var = c("RNA"))[, c(1, 3)]

# To make the relationship between Ct and RNA copies linear, we take
# log(RNA). Note that here log is actually ln and not log10
pcr.std$logRNA <- log10(pcr.std$RNA)

# perform the standard regression
std <- lm(logRNA ~ value, data = pcr.std)
summary(std)
```

```
##
## Call:
## lm(formula = logRNA ~ value, data = pcr.std)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1785 -0.1924  0.0933  0.2009  1.2962
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.088476   0.144127   70.00   <2e-16 ***
## value       -0.260465   0.005706  -45.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4196 on 73 degrees of freedom
##   (5 observations deleted due to missingness)
```

```
## Multiple R-squared:  0.9662, Adjusted R-squared:  0.9657
## F-statistic:  2084 on 1 and 73 DF,  p-value: < 2.2e-16
```

```
# Copy the Ct from the clinical data (so we don't need to change the name of
# the column) and use that to predict the log(RNA copies)
x1 <- data.frame(value = as.numeric(data$PCR1))
data$logRNA1 <- predict(std, x1, interval = "none") + log10(107.14)
x2 <- data.frame(value = as.numeric(data$PCR2))
data$logRNA2 <- predict(std, x2, interval = "none") + log10(107.14)

# We want to model the mean of the two PCRs (when 2 Ct values are available)
data$MlogRNA <- rowMeans(data[, c("logRNA1", "logRNA2")], na.rm = TRUE)

# Remove rows where we have no RNA measurement (some patients were only
# recorded as 'pos' or 'neg' or 'nil' or 'NA')
data <- data[!is.na(data$MlogRNA), ]
```

In this analysis, we are only interested in modeling the first detection of viremia. Positive patients were sampled longitudinally and we do not need the later timepoints.

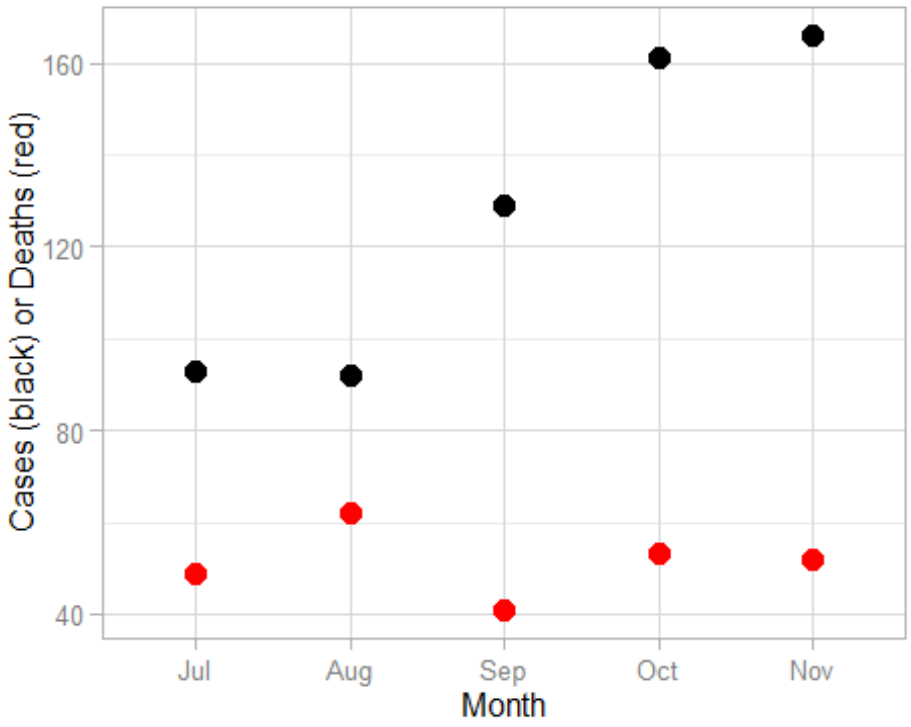
```
# Format the dates as dates instead of factors
data$Sampling <- as.Date(data$Sampling)
data$Admission <- as.Date(data$Admission)
data$Onset <- as.Date(data$Onset)
data$Outcome_Date <- as.Date(data$Outcome_Date)

# Take all the rows for each patient, order them by sampling date
# (Ascending) and keep the first row (the first tested sample)
data.first <- ldply(unique(data$Patient_ID), function(x) {
    temp <- data[data$Patient_ID == x, ]
    temp <- temp[order(temp$Sampling), ]
    if (is.na(temp[1, "Outcome"]))
        print(x)
    return(temp[1, ])
})

# Calculate the number of days between onset of symptoms (estimated by
# patients/doctors) and first sampling
data.first$SinceOnset <- as.numeric(data.first$Sampling - data.first$Onset)

# Keep only the rows where we were able to calculate SinceOnset. If it is
# NA, then one of the two dates is missing.
data.first <- data.first[!is.na(data.first$SinceOnset), ]

# Patient 460 appears to have an error in the sampling date (listed as April
# 1st 2014)
data.first <- data.first[data.first$Patient_ID != 460, ]

# If a regression is run over the Sampling date at the moment, its '0' value
# corresponds to the date 01Jan0000, which is meaningless in this context.
```

```r
# We subtract the first sampling date (01Jul2014) and start counting from
# there. This time count is divided by 10 so that its coefficient will not
# be too small, as the change in log(RNA) is not very large and this
# variable has a range of about 0-200.
data.first$Time0 <- as.numeric((data.first$Sampling -
min(data.first$Sampling)))/10

# Re-zero the time since onset to its mean. This will make possible
# interactions easier to interpret in the final model.
data.first$cOnset <- as.numeric(data.first$SinceOnset -
mean(data.first$SinceOnset))

# In the event we want to run a survival analysis: Codify survival as 0
# (survived) or 1 (Died)
data.first$event <- ifelse(data.first$Outcome == "Death", 1, 0)
# We would model the time difference as Day of Outcome - Day of first
# sampling. We do not use symptom onset as a start date because that date is
# estimated, whereas the day of first sampling is known.
data.first$time <- as.numeric(data.first$Outcome_Date - data.first$Sampling)

# We are not interested in the month of December, as it only has 2
# datapoints
data.first <- data.first[data.first$Sampling < as.Date("2014-12-01"), ]

head(data.first)
```

```
##   Patient_ID   Sampling Lab_number PCR1 PCR1.pos PCR2 PCR2.pos Sex
## 2          8 2014-07-01          1   19        1   NA       NA   M
## 3          9 2014-07-01          2   22        1   NA       NA   M
## 4          6 2014-07-02          6   26        1   NA       NA   F
## 5         10 2014-07-02          5   22        1   NA       NA   M
## 6         11 2014-07-02          4   23        1   NA       NA   M
## 7         13 2014-07-02          9   30        1   NA        1   M
##   Age_years  Admission      Onset Outcome_Date Outcome  logRNA1 logRNA2
## 2        35 2014-06-30 2014-06-27   2014-07-02   Death 7.169587      NA
## 3         3 2014-06-30 2014-06-28   2014-07-07   Death 6.388191      NA
## 4        50 2014-06-29 2014-06-20   2014-07-15   Cured 5.346330      NA
## 5        50 2014-07-01 2014-06-23   2014-07-03   Death 6.388191      NA
## 6        23 2014-07-01 2014-06-26   2014-07-03   Death 6.127726      NA
## 7         8 2014-07-02 2014-06-30   2014-07-07   Death 4.304469      NA
##    MlogRNA SinceOnset Time0     cOnset event time
## 2 7.169587          4   0.0 -4.2136223     1    1
## 3 6.388191          3   0.0 -5.2136223     1    6
## 4 5.346330         12   0.1  3.7863777     0   13
## 5 6.388191          9   0.1  0.7863777     1    1
## 6 6.127726          6   0.1 -2.2136223     1    1
## 7 4.304469          2   0.1 -6.2136223     1    5
```

The final number of cases examined is: 641 cases.

```
# graph the number of cases and deaths per month
data.first %>% dplyr::mutate(month = factor(format(Sampling, "%b"), levels =
c("Jul",
    "Aug", "Sep", "Oct", "Nov", "Dec"))) %>% group_by(month) %>%
dplyr::summarize(n = n(),
    death = sum(event)) %>% ggplot(aes(x = month, y = n)) + geom_point(size =
4,
    colour = "black") + geom_point(aes(y = death), size = 4, colour = "red")
+
    theme_light() + labs(x = "Month", y = "Cases (black) or Deaths (red)")
```



Model building

Main effects

```
regression.date <- lm(MlogRNA ~ Time0, data = data.first)
summary(regression.date)

##
## Call:
## lm(formula = MlogRNA ~ Time0, data = data.first)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5192 -1.3072  0.1048  1.2373  3.6193
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.80083    0.13472  43.057  < 2e-16 ***
## Time0       -0.10152    0.01396  -7.271 1.04e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.508 on 639 degrees of freedom
## Multiple R-squared:  0.07642,    Adjusted R-squared:  0.07497
## F-statistic: 52.87 on 1 and 639 DF,  p-value: 1.045e-12
```

```
regression.sex <- lm(MlogRNA ~ Sex, data = data.first)
summary(regression.sex)
```

```
##
## Call:
## lm(formula = MlogRNA ~ Sex, data = data.first)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0755 -1.4345  0.1338  1.3785  3.8194
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.93155    0.08788   56.12   <2e-16 ***
## SexM        -0.01854    0.12400   -0.15    0.881
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.57 on 639 degrees of freedom
## Multiple R-squared:  3.499e-05,  Adjusted R-squared:  -0.00153
## F-statistic: 0.02236 on 1 and 639 DF,  p-value: 0.8812
```

```
regression.age <- lm(MlogRNA ~ Age_years, data = data.first)
summary(regression.age)
```

```
##
## Call:
## lm(formula = MlogRNA ~ Age_years, data = data.first)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1151 -1.4326  0.1632  1.2902  3.2638
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.594564   0.130440  35.224  < 2e-16 ***
## Age_years   0.011771   0.004117   2.859  0.00439 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.56 on 635 degrees of freedom
```

```
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.01271,    Adjusted R-squared:  0.01116
## F-statistic: 8.175 on 1 and 635 DF,  p-value: 0.004386
```

```
regression.Onset <- lm(MlogRNA ~ cOnset, data = data.first)
summary(regression.Onset)
```

```
##
## Call:
## lm(formula = MlogRNA ~ cOnset, data = data.first)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3215 -1.2990  0.0607  1.2382  5.3635
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.922129   0.059034   83.38  < 2e-16 ***
## cOnset      -0.069713   0.008596   -8.11 2.59e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.495 on 639 degrees of freedom
## Multiple R-squared:  0.09332,    Adjusted R-squared:  0.0919
## F-statistic: 65.77 on 1 and 639 DF,  p-value: 2.592e-15
```

```
regression.Outcome <- lm(MlogRNA ~ Outcome, data = data.first)
summary(regression.Outcome)
```

```
##
## Call:
## lm(formula = MlogRNA ~ Outcome, data = data.first)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9242 -0.8326 -0.0512  0.7953  3.3349
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.09518    0.06121   66.91   <2e-16 ***
## OutcomeDeath  2.06280    0.09667   21.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.199 on 639 degrees of freedom
## Multiple R-squared:  0.4161, Adjusted R-squared:  0.4152
## F-statistic: 455.4 on 1 and 639 DF,  p-value: < 2.2e-16
```

Time0, Age, cOnset, and Outcome are significant predictors. Sex is not significant. While the time since the onset of symptoms may carry a significant level of error (as stated in the introduction), we believe that it remains a very important predictor. For cases of patients

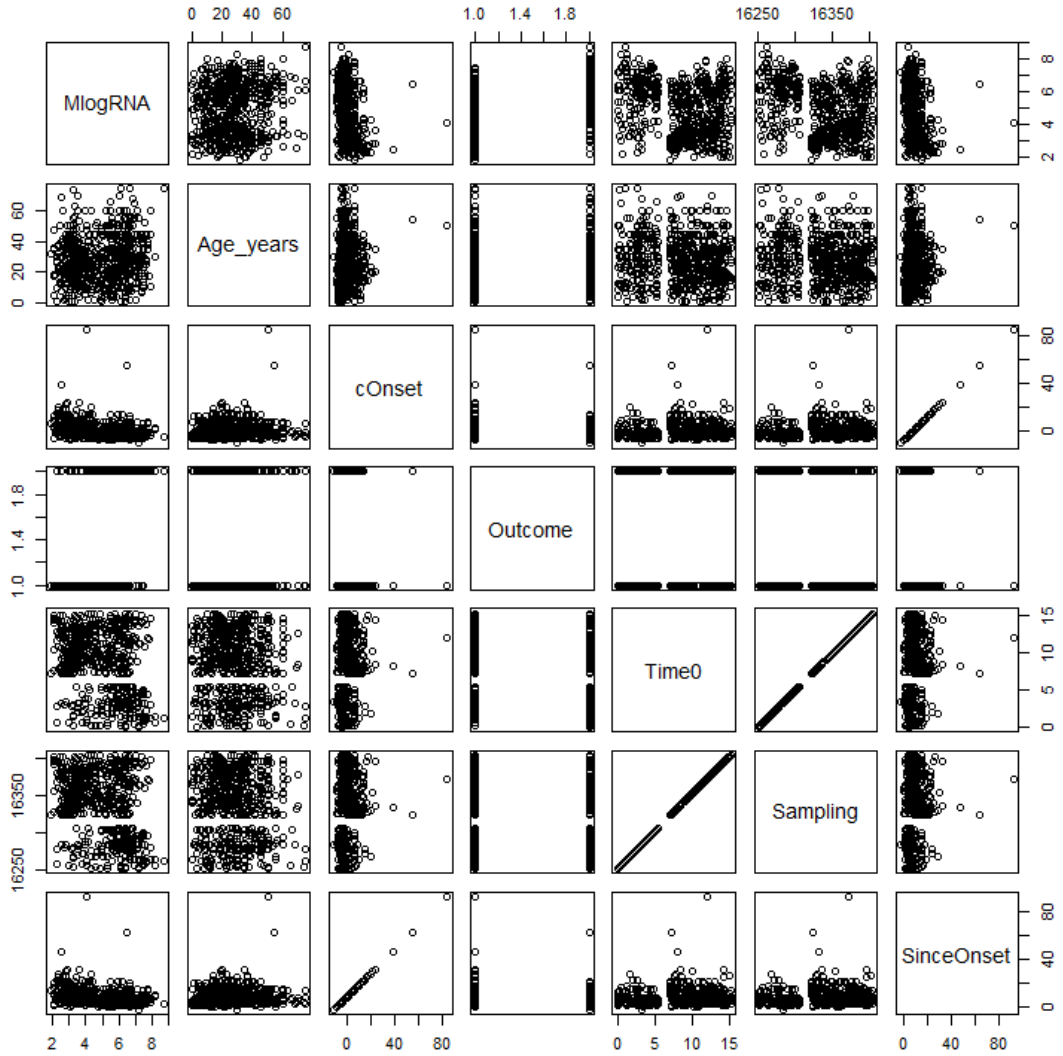
who succumbed to EBOV disease, the longer after onset, the higher the viremia should be; while for patients who recover, the viremia would decrease after too long.

Now we will work with a smaller dataframe.

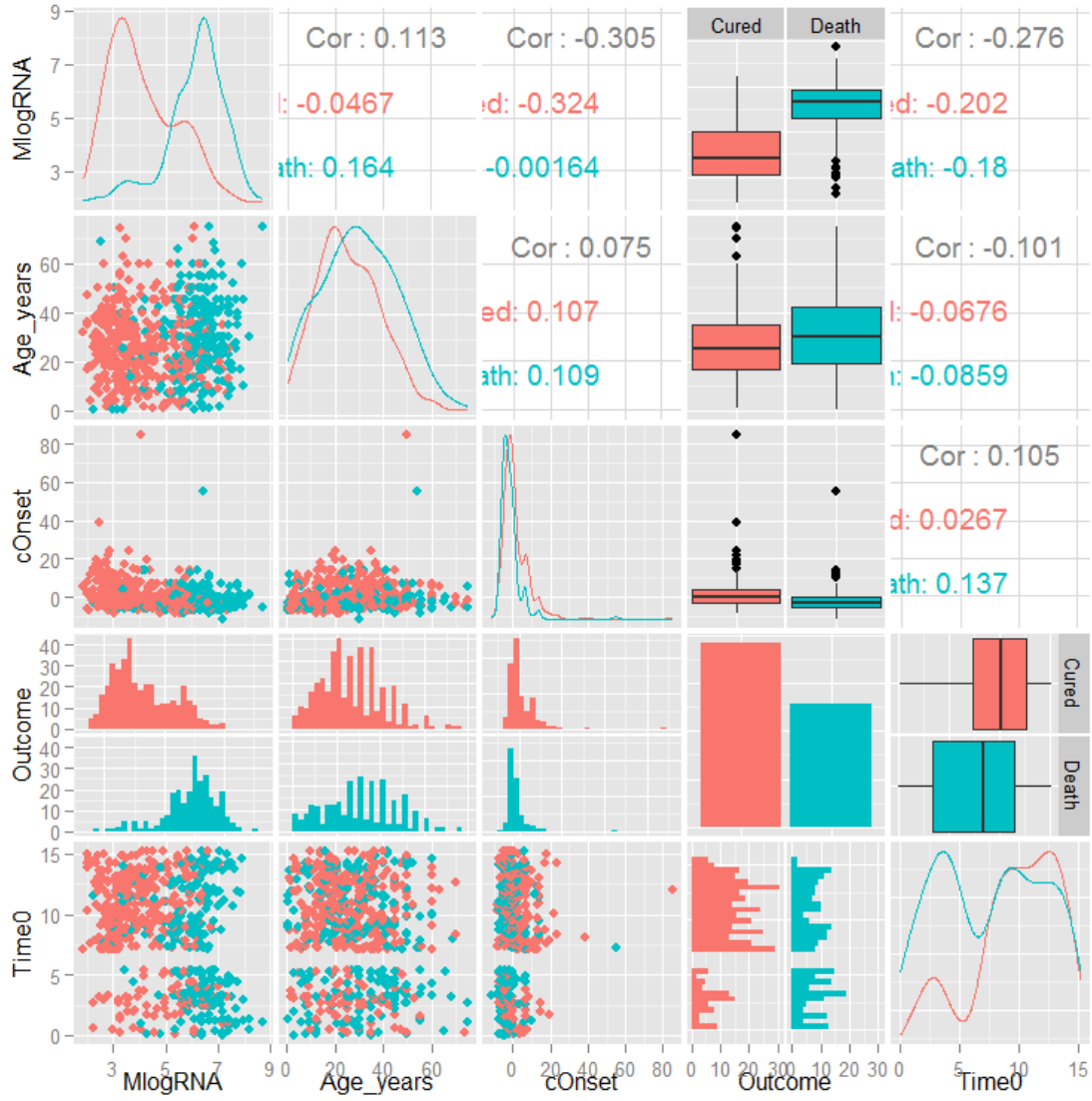
```
data.reg <- data.first[, c("MlogRNA", "Age_years", "cOnset", "Outcome",
"Time0",
    "Sampling", "SinceOnset")]
```

Look at the relationships between all variables:

```
plot(data.reg)
```

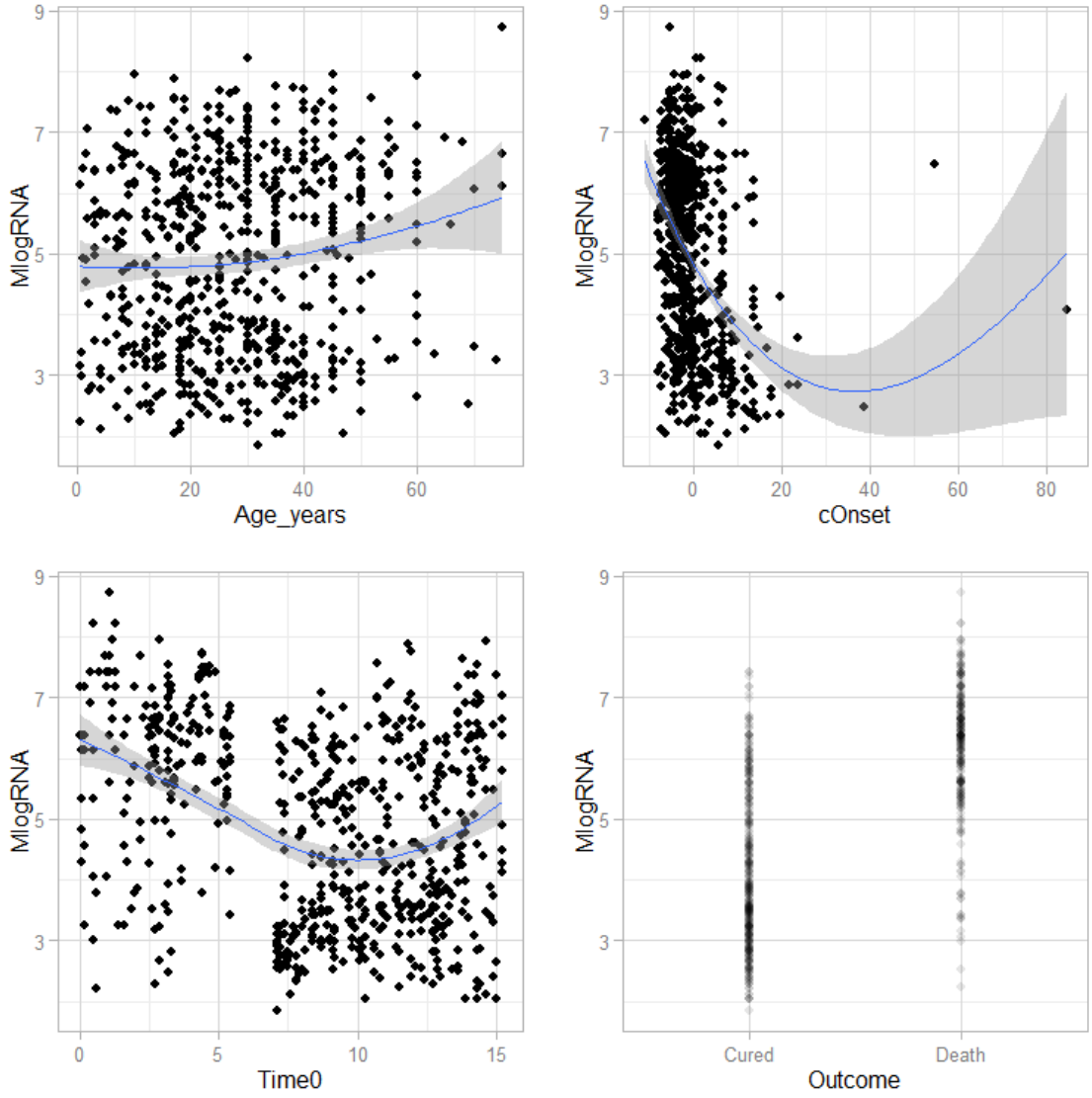



```
tmp <- data.table(data.reg)

ggpairs(data = tmp, diag = list(continuous = "density"), columns = c(1, 2, 3,
    4, 5), colour = "Outcome", axisLabels = "show")
```



Look specifically at the relationships between our chosen main effects and the outcome:

```
Age <- ggplot(data.reg, aes(x = Age_years, y = MlogRNA)) + geom_point() +
theme_light() +
    stat_smooth(span = 1)
Time0 <- ggplot(data.reg, aes(x = Time0, y = MlogRNA)) + geom_point() +
theme_light() +
    stat_smooth(span = 1)
cOnset <- ggplot(data.reg, aes(x = cOnset, y = MlogRNA)) + geom_point() +
theme_light() +
    stat_smooth(span = 1)
Outcome <- ggplot(data.reg, aes(x = Outcome, y = MlogRNA)) + geom_point(alpha
= 0.1) +
    theme_light() + stat_smooth(span = 1)
multiplot(Age, Time0, cOnset, Outcome, cols = 2)
```



Time0 has a non-linear relationship with MlogRNA, we will try to incorporate Time0^2 in the model. cOnset looks non-linear, but this is due to 3 strange observations; we will look at the model with and without those observations.

First model with all main effects together:

```
data.reg$Time0Sq <- data.reg$Time0^2
model.ME <- lm(MlogRNA ~ Age_years + cOnset + Outcome + Time0 + Time0Sq, data
= data.reg)
summary(model.ME)

##
## Call:
## lm(formula = MlogRNA ~ Age_years + cOnset + Outcome + Time0 +
##     Time0Sq, data = data.reg)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5604 -0.7403 -0.0287  0.7496  2.9821
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.183902   0.185316  27.973  < 2e-16 ***
## Age_years     0.003704   0.003019   1.227     0.22
## cOnset       -0.035325   0.006882  -5.133 3.81e-07 ***
## OutcomeDeath  1.817871   0.095591  19.017  < 2e-16 ***
## Time0        -0.295331   0.043580  -6.777 2.82e-11 ***
## Time0Sq       0.015680   0.002697   5.813 9.74e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.124 on 631 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.4908, Adjusted R-squared:  0.4868
## F-statistic: 121.6 on 5 and 631 DF,  p-value: < 2.2e-16
```

Age is not significant anymore, drop it from the model:

```
model.ME2 <- lm(MlogRNA ~ cOnset + Outcome + Time0 + Time0Sq, data =
data.reg)
summary(model.ME2)

##
## Call:
## lm(formula = MlogRNA ~ cOnset + Outcome + Time0 + Time0Sq, data =
data.reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6692 -0.7384 -0.0515  0.7284  2.9896
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.287410   0.162795  32.479  < 2e-16 ***
## cOnset       -0.034340   0.006653  -5.162 3.27e-07 ***
## OutcomeDeath  1.836132   0.094222  19.487  < 2e-16 ***
## Time0        -0.294680   0.043363  -6.796 2.49e-11 ***
## Time0Sq       0.015561   0.002684   5.798 1.06e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.122 on 636 degrees of freedom
## Multiple R-squared:  0.4918, Adjusted R-squared:  0.4886
## F-statistic: 153.9 on 4 and 636 DF,  p-value: < 2.2e-16
```

All main effects left are significant.

Interactions

Add interactions to the model, keep the significant ones. Time since onset of symptoms with Outcome:

```
model.Int1 <- lm(MlogRNA ~ cOnset * Outcome + Time0 + Time0Sq, data.reg)
summary(model.Int1)

##
## Call:
## lm(formula = MlogRNA ~ cOnset * Outcome + Time0 + Time0Sq, data =
data.reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5927 -0.6971 -0.0009  0.7140  4.2859
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.333047   0.161294  33.064  < 2e-16 ***
## cOnset              -0.049750   0.007613  -6.535 1.31e-10 ***
## OutcomeDeath         1.890759   0.094109  20.091  < 2e-16 ***
## Time0               -0.297546   0.042863  -6.942 9.58e-12 ***
## Time0Sq              0.015554   0.002652   5.864 7.26e-09 ***
## cOnset:OutcomeDeath  0.059878   0.014912   4.015 6.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.109 on 635 degrees of freedom
## Multiple R-squared:  0.5044, Adjusted R-squared:  0.5005
## F-statistic: 129.2 on 5 and 635 DF,  p-value: < 2.2e-16
```

Significant, keep it in the model. Time since onset of symptoms with time since beginning of dataset:

```
model.Int2 <- lm(MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Time0Sq,
data.reg)
summary(model.Int2)

##
## Call:
## lm(formula = MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Time0Sq,
##     data = data.reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5246 -0.7045 -0.0489  0.7106  3.2345
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.260172   0.162168  32.437  < 2e-16 ***
```

```
## cOnset               -0.106209   0.020432  -5.198 2.72e-07 ***
## OutcomeDeath          1.885746   0.093548  20.158  < 2e-16 ***
## Time0                -0.284701   0.042819  -6.649 6.37e-11 ***
## Time0Sq               0.015082   0.002641   5.711 1.73e-08 ***
## cOnset:OutcomeDeath   0.074008   0.015563   4.755 2.45e-06 ***
## cOnset:Time0          0.005725   0.001925   2.975  0.00304 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.102 on 634 degrees of freedom
## Multiple R-squared:  0.5112, Adjusted R-squared:  0.5066
## F-statistic: 110.5 on 6 and 634 DF,  p-value: < 2.2e-16
```

Significant, keep it in the model. Time since the beginning of the dataset and outcome:

```
model.Int3 <- lm(MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome *
Time0,
    data.reg)
summary(model.Int3)

##
## Call:
## lm(formula = MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome *
##     Time0, data = data.reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6881 -0.7606 -0.0097  0.7517  3.3788
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.736718   0.148293  31.942  < 2e-16 ***
## cOnset               -0.122094   0.021127  -5.779 1.18e-08 ***
## OutcomeDeath          1.635720   0.209645   7.802 2.51e-14 ***
## Time0                -0.061864   0.014570  -4.246 2.50e-05 ***
## cOnset:OutcomeDeath   0.074873   0.015939   4.697 3.23e-06 ***
## cOnset:Time0          0.006928   0.002002   3.461 0.000575 ***
## OutcomeDeath:Time0    0.031750   0.021893   1.450 0.147490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.128 on 634 degrees of freedom
## Multiple R-squared:  0.4878, Adjusted R-squared:  0.4829
## F-statistic: 100.6 on 6 and 634 DF,  p-value: < 2.2e-16
```

Not significant, remove from the model. Time since onset of symptoms and time since the beginning of the dataset squared:

```
model.Int4 <- lm(MlogRNA ~ cOnset * Outcome + cOnset * Time0 + cOnset *
Time0Sq,
```

```
    data.reg)
summary(model.Int4)

##
## Call:
## lm(formula = MlogRNA ~ cOnset * Outcome + cOnset * Time0 + cOnset *
##     Time0Sq, data = data.reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5232 -0.6977 -0.0562  0.7111  3.2455
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.2549603  0.1636082  32.119  < 2e-16 ***
## cOnset              -0.1120854  0.0310690  -3.608 0.000333 ***
## OutcomeDeath         1.8846847  0.0937128  20.111  < 2e-16 ***
## Time0               -0.2840048  0.0429399  -6.614 7.96e-11 ***
## Time0Sq              0.0150555  0.0026450   5.692 1.92e-08 ***
## cOnset:OutcomeDeath  0.0735810  0.0156673   4.696 3.25e-06 ***
## cOnset:Time0         0.0075970  0.0076960   0.987 0.323954
## cOnset:Time0Sq      -0.0001160  0.0004618  -0.251 0.801723
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.103 on 633 degrees of freedom
## Multiple R-squared:  0.5113, Adjusted R-squared:  0.5059
## F-statistic: 94.59 on 7 and 633 DF,  p-value: < 2.2e-16
```

Not significant, remove from the model. Time since the beginning of the dataset squared and outcome:

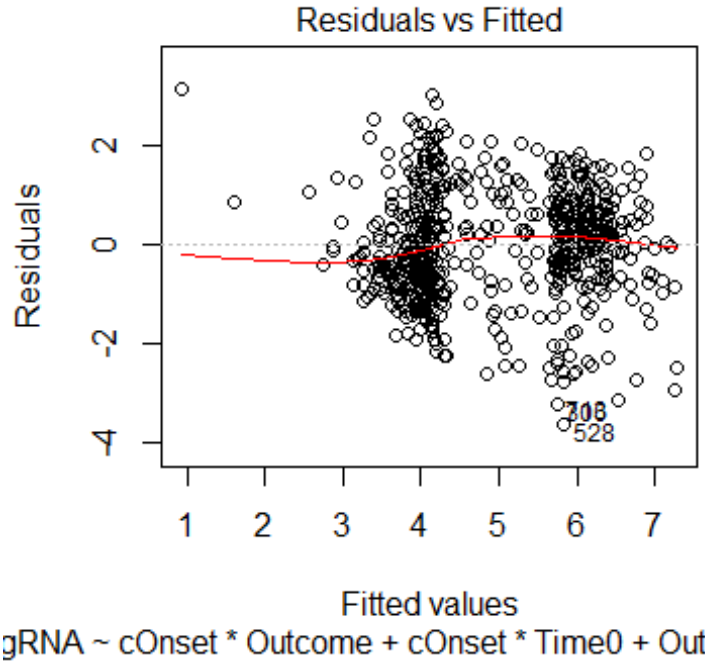
```
model.Int5 <- lm(MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome *
Time0Sq,
    data.reg)
summary(model.Int5)

##
## Call:
## lm(formula = MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome *
##     Time0Sq, data = data.reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6046 -0.6905 -0.0335  0.6612  3.1395
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          5.420252   0.176871  30.645  < 2e-16 ***
## cOnset              -0.113468   0.020626  -5.501 5.48e-08 ***
## OutcomeDeath         1.614212   0.153335  10.527  < 2e-16 ***
```

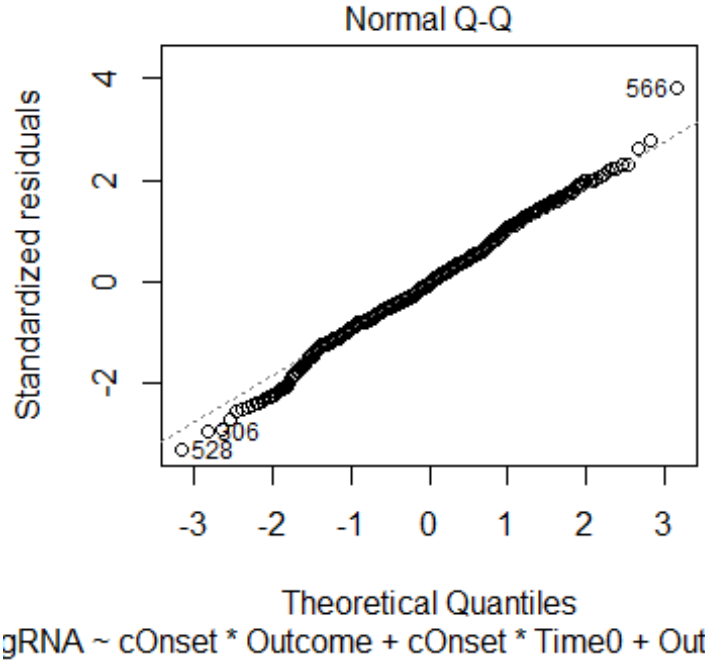
```
## Time0                -0.294968   0.042932  -6.871 1.53e-11 ***
## Time0Sq               0.014463   0.002647   5.463 6.72e-08 ***
## cOnset:OutcomeDeath   0.073447   0.015517   4.733 2.73e-06 ***
## cOnset:Time0          0.006475   0.001948   3.324 0.000938 ***
## OutcomeDeath:Time0Sq  0.002979   0.001335   2.231 0.026039 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.098 on 633 degrees of freedom
## Multiple R-squared:  0.515,  Adjusted R-squared:  0.5097
## F-statistic: 96.03 on 7 and 633 DF,  p-value: < 2.2e-16
```

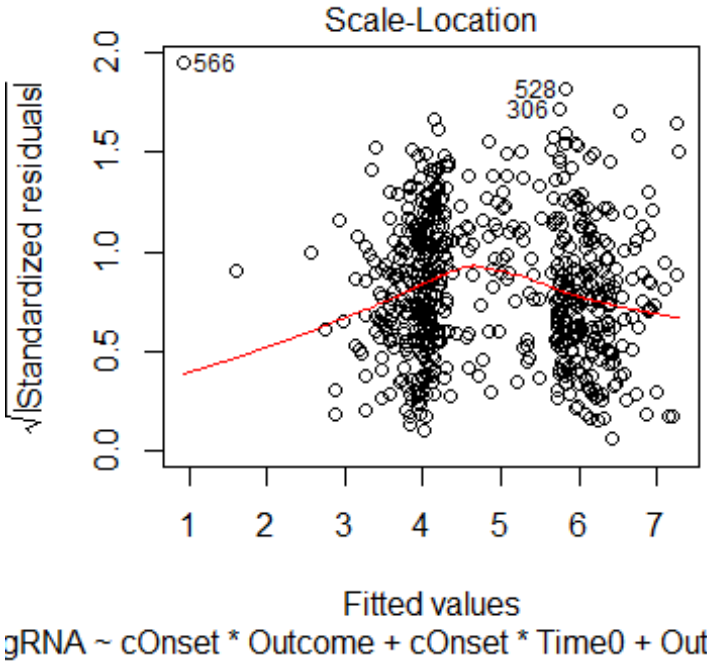
Significant at 0.1,keep in the model.

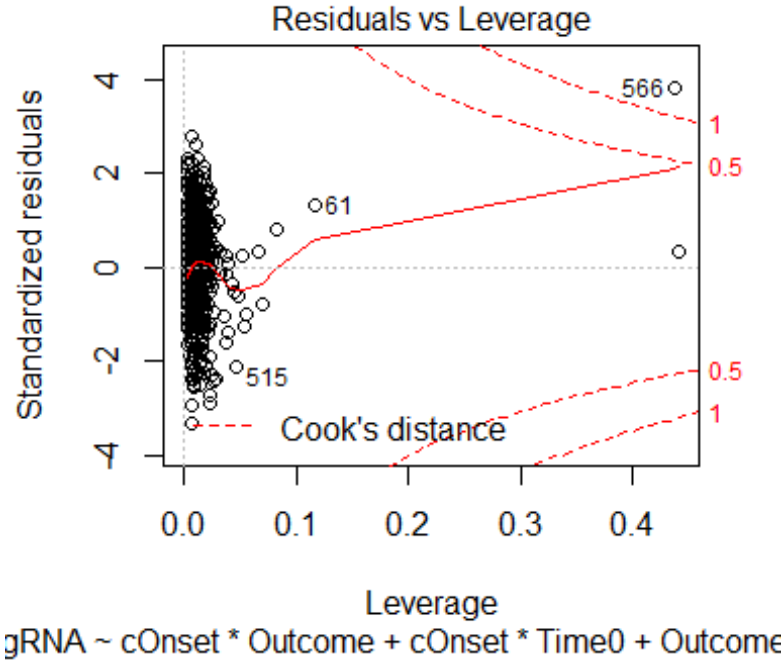
Final model:

```
model <- lm(MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome * Time0Sq,
    data.reg)
summary(model)

##
## Call:
## lm(formula = MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome *
##     Time0Sq, data = data.reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6046 -0.6905 -0.0335  0.6612  3.1395
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.420252   0.176871  30.645  < 2e-16 ***
## cOnset               -0.113468   0.020626  -5.501 5.48e-08 ***
## OutcomeDeath          1.614212   0.153335  10.527  < 2e-16 ***
## Time0                -0.294968   0.042932  -6.871 1.53e-11 ***
## Time0Sq               0.014463   0.002647   5.463 6.72e-08 ***
## cOnset:OutcomeDeath   0.073447   0.015517   4.733 2.73e-06 ***
## cOnset:Time0          0.006475   0.001948   3.324 0.000938 ***
## OutcomeDeath:Time0Sq  0.002979   0.001335   2.231 0.026039 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.098 on 633 degrees of freedom
## Multiple R-squared:  0.515,  Adjusted R-squared:  0.5097
## F-statistic: 96.03 on 7 and 633 DF,  p-value: < 2.2e-16

plot(model)
```


Residuals vs Fitted
Residuals
Fitted values
gRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome
708
528

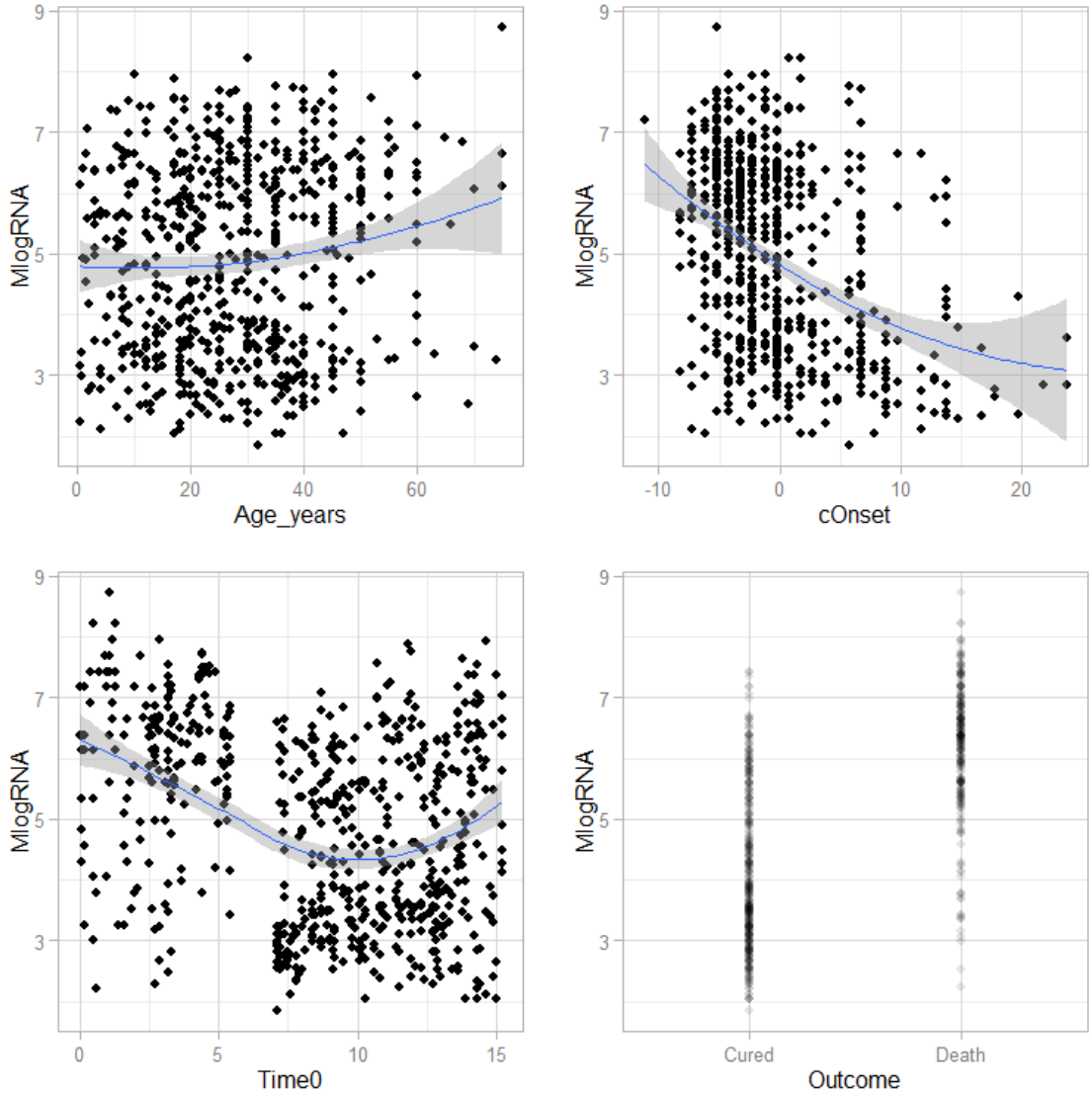

Normal Q-Q
Standardized residuals
Theoretical Quantiles
gRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome
566
306
528






```r
# Remove the three extreme Times from onset of symptoms
data.reg.mod <- data.reg[data.reg$cOnset < 30, ]
modelFinal <- lm(MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome *
Time0Sq,
```

```
    data.reg.mod)
summary(modelFinal)

##
## Call:
## lm(formula = MlogRNA ~ cOnset * Outcome + cOnset * Time0 + Outcome *
##     Time0Sq, data = data.reg.mod)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6403 -0.6715  0.0256  0.6736  2.9610
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.428402   0.174996  31.020  < 2e-16 ***
## cOnset               -0.110007   0.020957  -5.249 2.09e-07 ***
## OutcomeDeath          1.620576   0.156060  10.384  < 2e-16 ***
## Time0                -0.290123   0.042483  -6.829 2.02e-11 ***
## Time0Sq               0.014057   0.002621   5.364 1.15e-07 ***
## cOnset:OutcomeDeath   0.089056   0.019501   4.567 5.96e-06 ***
## cOnset:Time0          0.003478   0.002103   1.654   0.0987 .
## OutcomeDeath:Time0Sq  0.002749   0.001330   2.067   0.0392 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.086 on 630 degrees of freedom
## Multiple R-squared:  0.525,  Adjusted R-squared:  0.5197
## F-statistic: 99.48 on 7 and 630 DF,  p-value: < 2.2e-16

Age <- ggplot(data.reg.mod, aes(x = Age_years, y = MlogRNA)) + geom_point() +
    theme_light() + stat_smooth(span = 1)
Time0 <- ggplot(data.reg.mod, aes(x = Time0, y = MlogRNA)) + geom_point() +
    theme_light() + stat_smooth(span = 1)
cOnset <- ggplot(data.reg.mod, aes(x = cOnset, y = MlogRNA)) + geom_point() +
    theme_light() + stat_smooth(span = 1)
Outcome <- ggplot(data.reg.mod, aes(x = Outcome, y = MlogRNA)) +
geom_point(alpha = 0.1) +
    theme_light() + stat_smooth(method = "lm")
multiplot(Age, Time0, cOnset, Outcome, cols = 2)
```



The three extreme Onset times do not have a large impact on the regression.

Graphical summaries of the model

Create the graphing datasets

```
# Data to graph the effect of Sampling date for average time since onset
# (cOnset == 0) We only need 3 predictors from data.reg
Preds.Sampling <- data.reg[, c("Sampling", "Time0", "Outcome", "Time0Sq")]
# We do not need repeated values
Preds.Sampling <- unique(Preds.Sampling)
# set cOnset to its mean (0).
Preds.Sampling$cOnset <- 0
# The predict function returns the expected values and their 95% confidence
# interval
pred.vals <- data.frame(predict(model, Preds.Sampling, interval =
```

```
"confidence"))
names(pred.vals) <- c("MlogRNA", "L95", "U95")
# Attach the predictions to the predictors
Preds.Sampling <- cbind(Preds.Sampling, pred.vals)


# Data to graph the effect of time since onset for the middle of the dataset
# (at the mean(Time0)) We only need 3 predictors from data.reg
Preds.Onset <- data.reg[, c("SinceOnset", "cOnset", "Outcome")]
# We do not need repeated values
Preds.Onset <- unique(Preds.Onset[, c("SinceOnset", "cOnset", "Outcome")])
# set Time0 to its mean.
Preds.Onset$Time0 <- mean(data.reg$Time0)
Preds.Onset$Time0Sq <- mean(data.reg$Time0)^2
# The predict function returns the expected values and their 95% confidence
# interval
pred.vals <- data.frame(predict(model, Preds.Onset, interval = "confidence"))
names(pred.vals) <- c("MlogRNA", "L95", "U95")
# Attach the predictions to the predictors
Preds.Onset <- cbind(Preds.Onset, pred.vals)
```

Graphs

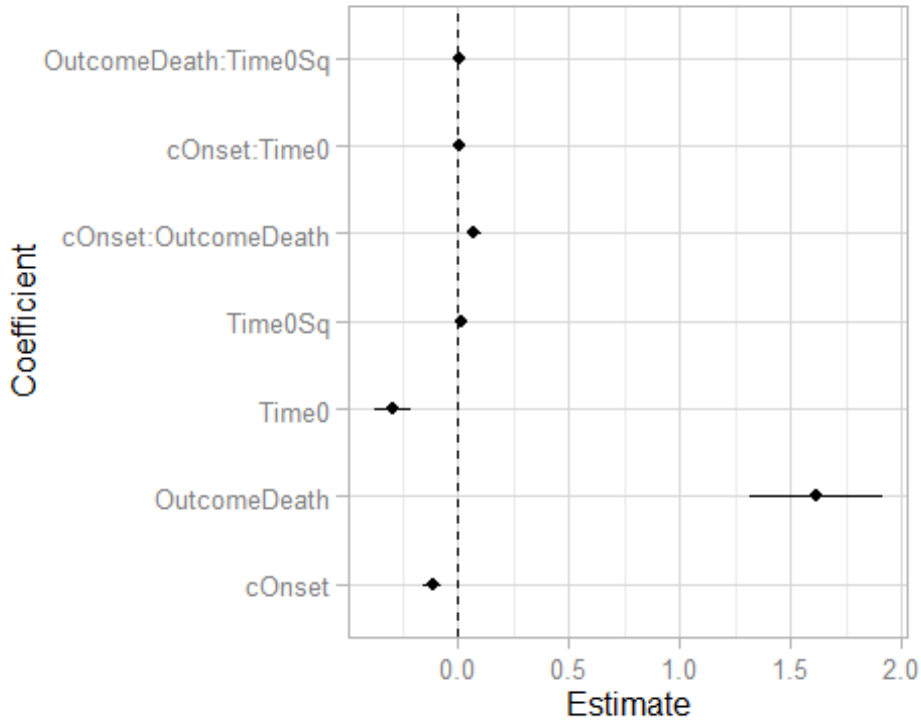
Graphs of the regression coefficients

```
# The summary function calculates the standard errors. Obtain the table
# containing the estimates and errors
reg.coeffs <- summary(model)$coefficients
# Get all the values we need for our dataset: Names of the predictors
coef.names <- rownames(reg.coeffs)
# Estimates of the coefficients
coef.est <- reg.coeffs[, 1]
# Standard errors of the coefficients
coef.err <- reg.coeffs[, 2]
# 95% CI of the coefficients
coef.L95 <- coef.est - 1.96 * coef.err
coef.U95 <- coef.est + 1.96 * coef.err
# Assemble in a dataframe
coef.data <- data_frame(Coefficient = coef.names, Estimate = coef.est, U95 =
coef.U95,
    L95 = coef.L95)
# Reorder the coefficients in the same order as the regression results
coef.data$Coefficient <- factor(coef.data$Coefficient, levels =
c("(Intercept)",
    "cOnset", "OutcomeDeath", "Time0", "Time0Sq", "cOnset:OutcomeDeath",
"cOnset:Time0",
    "OutcomeDeath:Time0Sq"))
# drop intercept
coef.data <- coef.data[-1, ]
```

```
# show the data
coef.data
```

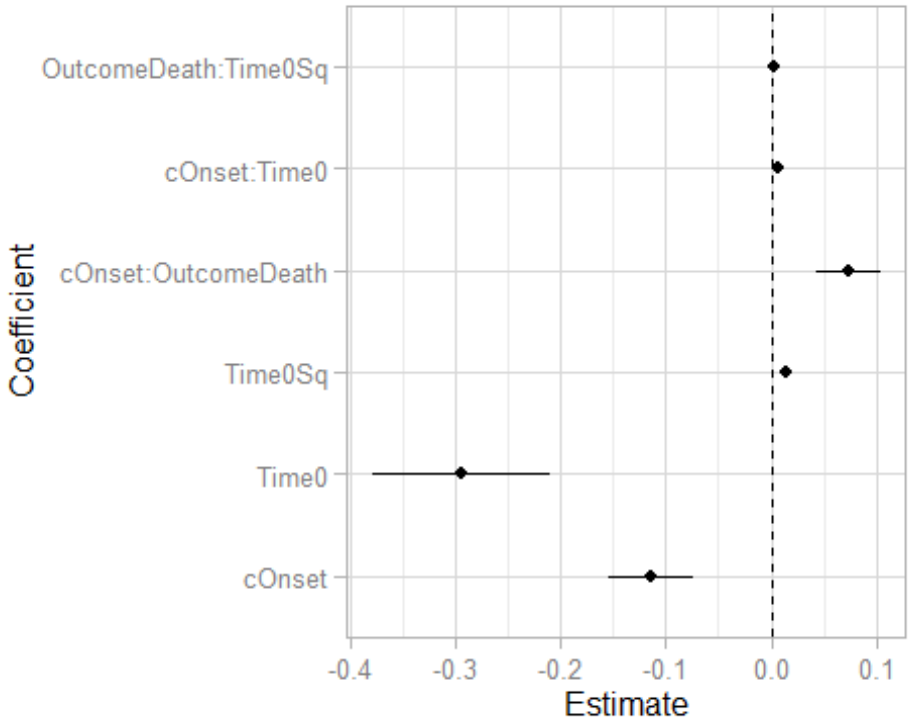
```
## Source: local data frame [7 x 4]
##
##            Coefficient     Estimate          U95           L95
## 1              cOnset -0.113468034 -0.073040352 -0.1538957166
## 2         OutcomeDeath  1.614212270  1.914749096  1.3136754428
## 3                Time0 -0.294967732 -0.210820450 -0.3791150147
## 4              Time0Sq  0.014462898  0.019651495  0.0092743000
## 5  cOnset:OutcomeDeath  0.073447452  0.103860054  0.0430348497
## 6         cOnset:Time0  0.006474926  0.010292592  0.0026572606
## 7 OutcomeDeath:Time0Sq  0.002978920  0.005596141  0.0003616982
```

```
# All coefficients
ggplot(coef.data, aes(y = Estimate, x = Coefficient)) +
geom_pointrange(aes(ymin = L95,
    ymax = U95)) + geom_hline(yintercept = 0, linetype = 2) + theme_light() +
    coord_flip()
```




```
# save a copy
pdf("../Results/Marc/Makona-AllCoefs.pdf", width = 6.5, height = 2.5,
useDingbats = FALSE)
ggplot(coef.data, aes(y = Estimate, x = Coefficient)) +
geom_pointrange(aes(ymin = L95,
    ymax = U95)) + geom_hline(yintercept = 0, linetype = 2) + theme_light() +
```

```
    coord_flip()
dev.off()

## png
##   2

# focus on the coefficients near 0
ggplot(coef.data[c(1, 3:7), ], aes(y = Estimate, x = Coefficient)) +
geom_pointrange(aes(ymin = L95,
    ymax = U95)) + geom_hline(yintercept = 0, linetype = 2) + theme_light() +
    coord_flip()
```



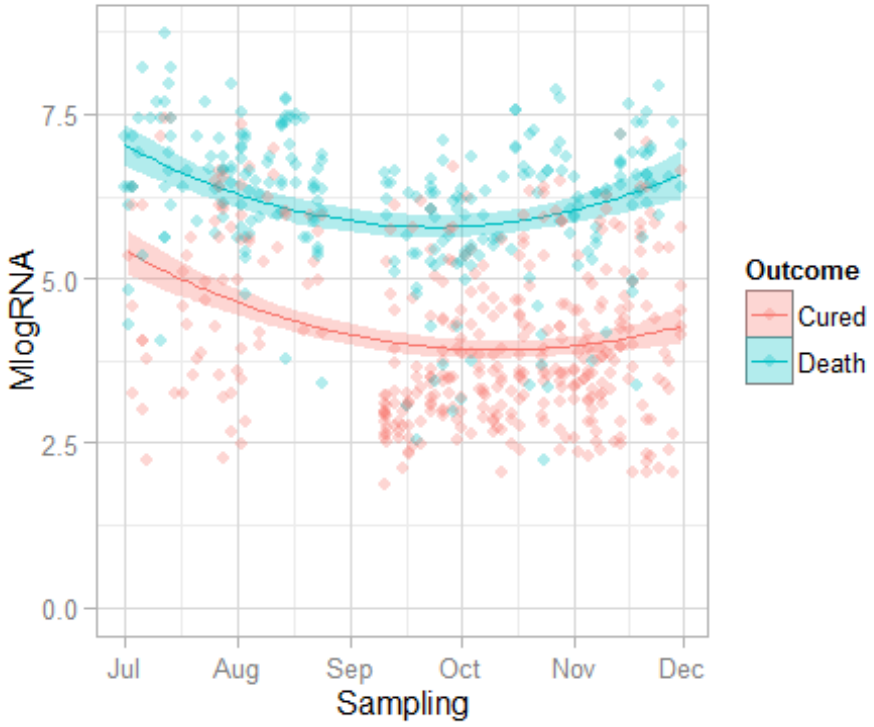

```
# save a copy
pdf("../Results/Marc/Makona-smallCoefs.pdf", width = 6.5, height = 2.5,
useDingbats = FALSE)
ggplot(coef.data[c(1, 3:7), ], aes(y = Estimate, x = Coefficient)) +
geom_pointrange(aes(ymin = L95,
    ymax = U95)) + geom_hline(yintercept = 0, linetype = 2) + theme_light() +
    coord_flip()
dev.off()

## png
##   2
```

Regression graphs

Effect of sampling date

(Time since data collection began):

```
ggplot(data.reg, aes(x = Sampling, y = MlogRNA, colour = Outcome, fill =
Outcome)) +
    geom_ribbon(data = Preds.Sampling, aes(ymin = L95, ymax = U95), linetype
= 0,
        alpha = 0.3) + geom_line(data = Preds.Sampling) + geom_point(alpha =
0.3) +
    ylim(0, NA) + theme_light()
```

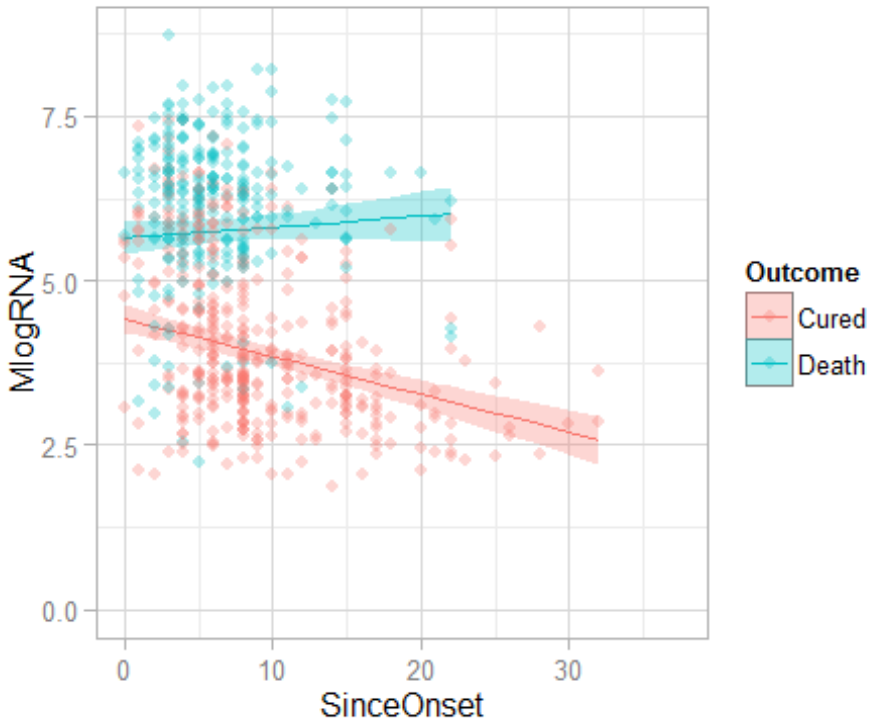



```
ggsave("../Results/Marc/Makona-LogRNASampling.pdf")

## Saving 5 x 4 in image
```

Effect of time since onset:

```
ggplot(data.reg, aes(x = SinceOnset, y = MlogRNA, colour = Outcome, fill =
Outcome)) +
    geom_ribbon(data = Preds.Onset, aes(ymin = L95, ymax = U95), linetype =
0,
        alpha = 0.3) + geom_line(data = Preds.Onset) + geom_point(alpha =
0.3) +
    xlim(0, 37.5) + ylim(0, NA) + theme_light()
```

```
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_point).
```




```
ggsave("../Results/Marc/Makona-LogRNAOnset.pdf")
## Saving 5 x 4 in image
## Warning: Removed 4 rows containing missing values (geom_path).
## Warning: Removed 4 rows containing missing values (geom_point).
```