Code ▾

Predicting COVID-19 with Multiple Linear Regression

Scott Dolan

12/6/2020

Load data and select desired desired data

Hide

```
rawdata <- read_csv('PA-COVID-19-Data-Merged.csv')
```

```
## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   County = col_character(),
##   daily.death.count = col_double(),
##   population.2018 = col_double(),
##   patients.hospitalized = col_double(),
##   patients.on.ventilators = col_double(),
##   airborn.isolation.beds = col_double(),
##   adult.ICU.beds = col_double(),
##   new.cases = col_double(),
##   month = col_character(),
##   region = col_character()
## )
```

Hide

```
berks.dat <- rawdata %>% dplyr::filter(.,County == 'Berks' & Date >= '2020-06-01')
date_as_factor.berks <- data.frame(Day=c(1:length(unique(berks.dat$Date))),Date=uniqu
e(berks.dat$Date))
berks.dat <- berks.dat %>% left_join(.,date_as_factor.berks,by=c("Date"))
berks.model.dat <- berks.dat %>% dplyr::filter(.,Date <= '2020-12-15')
berks.verify.dat <- berks.dat %>% dplyr::filter(.,Date > '2020-12-15')


philly.dat <- rawdata %>% dplyr::filter(.,County == 'Philadelphia' & Date >= '2020-06
-01')
date_as_factor.philly <- data.frame(Day=c(1:length(unique(philly.dat$Date))),Date=uni
que(philly.dat$Date))
philly.dat <- philly.dat %>% left_join(.,date_as_factor.philly,by=c("Date"))
philly.model.dat <- philly.dat %>% dplyr::filter(.,Date <= '2020-12-15')
philly.verify.dat <- philly.dat %>% dplyr::filter(.,Date > '2020-12-15')
```

Berks County model 1 using linear regression and power

transformation

Hide

```
# DV  : # of New Cases
# IVs : Day

getMod = function(df){
  p = seq(0,4,0.01)
  Rsquare_p = c(0,0,0,0)
  p.value <- 1
  model = list(NULL)
  for(i in c(2:length(p))){

    modDat <- df %>%  dplyr::mutate(., Xdat = (Day^p[i]-1)/p[i]) %>% drop_na()
    fit <- modDat %>% lm(new.cases ~ Xdat,.)
    sumFit <- summary(fit)
    f <- sumFit$fstatistic
    p.value.fit <- pf(f[1],f[2],f[3],lower.tail=F)
    if(sumFit$adj.r.squared > Rsquare_p[2] & p.value.fit < p.value){
      Rsquare_p <- c(sumFit$r.squared,sumFit$adj.r.squared,p[i],i)
      model <- fit
      p.value <- p.value.fit
      model.sum <- sumFit
    }
  }
  if(is_empty(model)==TRUE){
      print('Search higher power')
      Rsquare_p <- c(sumFit$r.squared,sumFit$adj.r.squared,p[i],i)
      model <- fit
      p.value <- p.value.fit
      model.sum <- sumFit
  }

  return(list(model,Rsquare_p,modDat$Day,p.value,model.sum))
}
berks.model1 <- getMod(berks.model.dat)
```

Explore model 1 for Berks County using linear regression (unweighted)

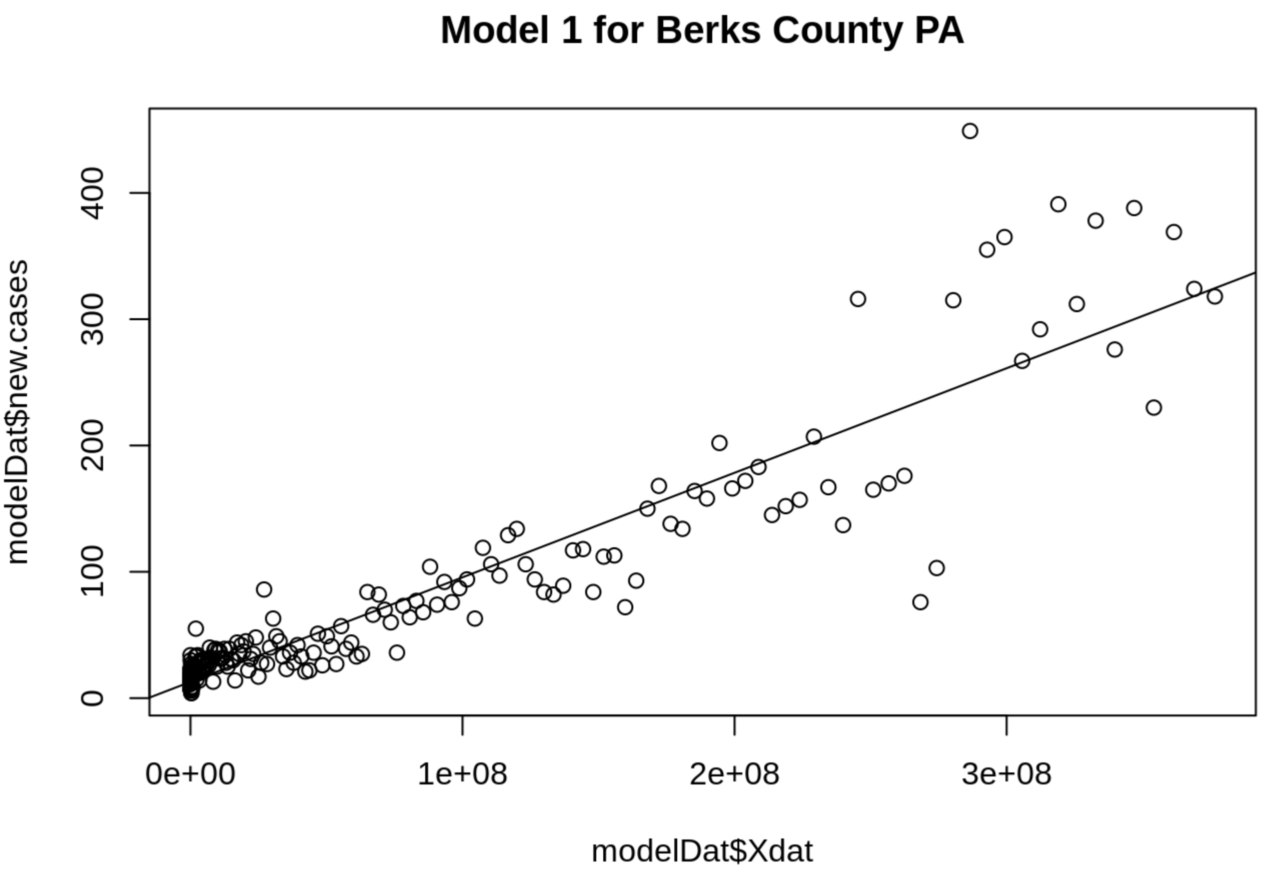
Hide

```
summary(berks.model1[[1]])
```

```
##
## Call:
## lm(formula = new.cases ~ Xdat, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -158.986   -9.925    1.074   10.964  198.920
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.291e+01  3.055e+00   4.225 3.67e-05 ***
## Xdat        8.277e-07  2.407e-08  34.388  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.28 on 195 degrees of freedom
## Multiple R-squared:  0.8584, Adjusted R-squared:  0.8577
## F-statistic:  1183 on 1 and 195 DF,  p-value: < 2.2e-16
```

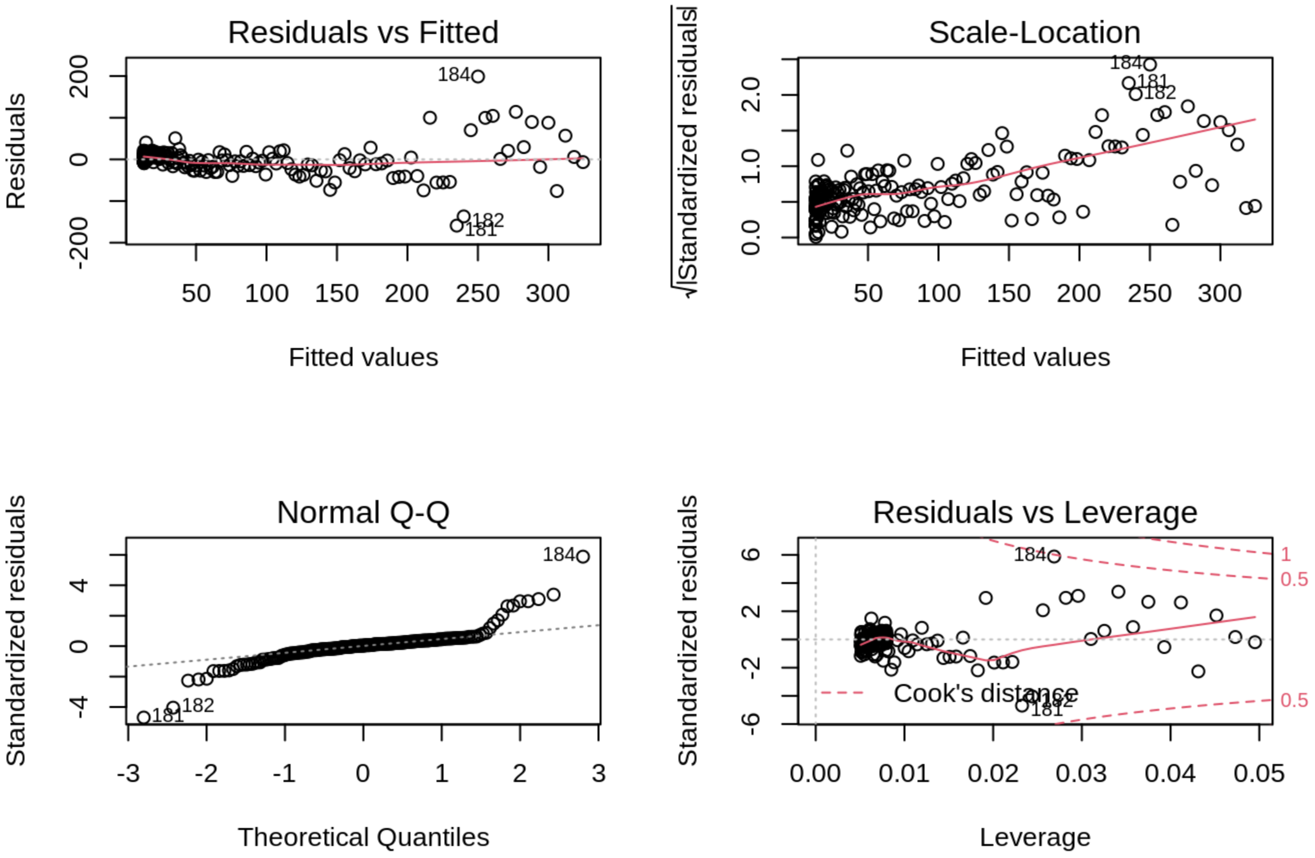
Hide

```
model1 = berks.model1[[1]]
modelDat <- data.frame(model1$model)
power <- berks.model1[[2]][[3]]
plot(modelDat$Xdat,modelDat$new.cases,main = 'Model 1 for Berks County PA')
abline(model1)
```



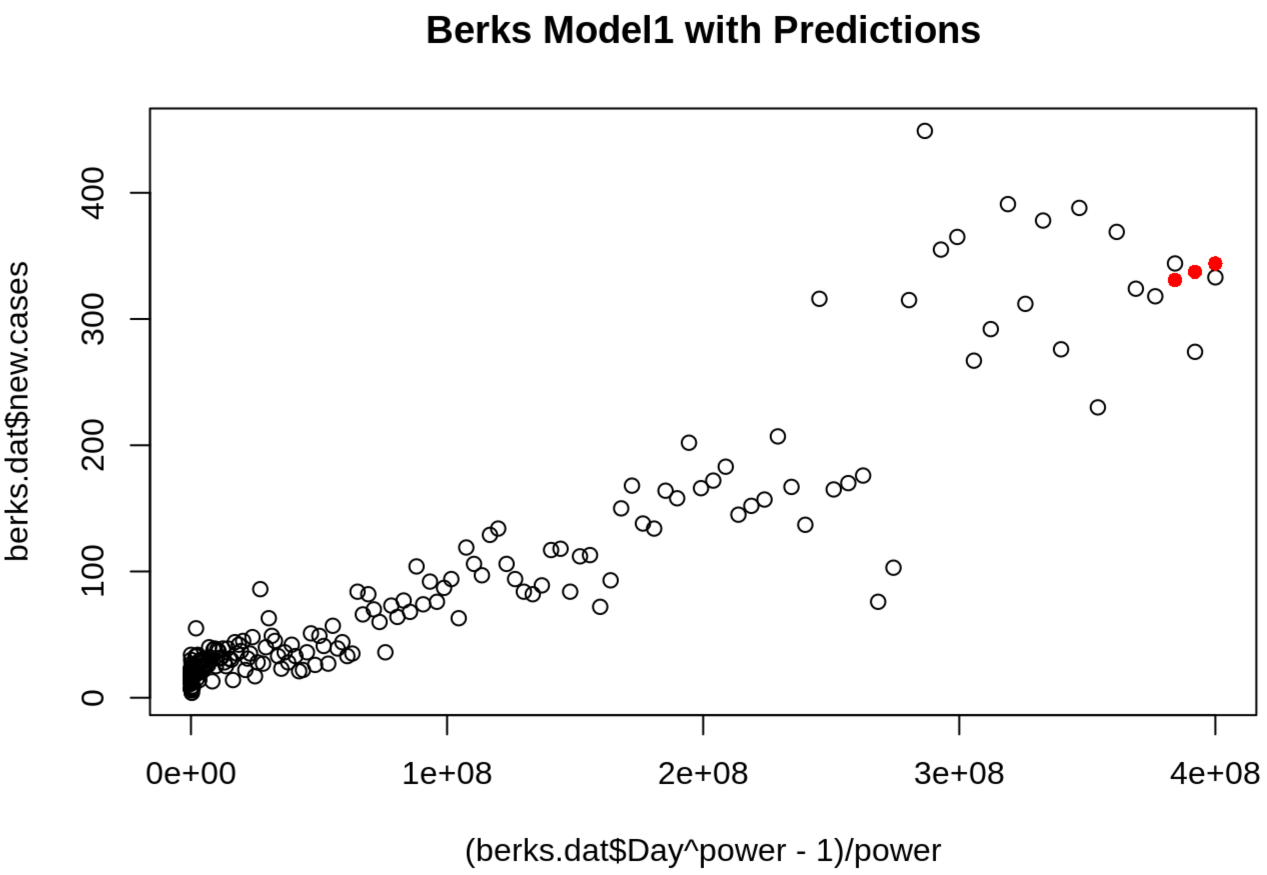
Hide

```
# get summary and plots of fit
layout(matrix(1:4,2,2))
plot(model1)
```



Hide

```
par(mfrow=c(1,1))

# Make predictions
Yhat <- function(B0,B1,day,p){B0+B1*(day^p-1)/p}

y.int <- model1$coefficients[1]
slop <- model1$coefficients[2]
berks.prediction <- Yhat(y.int,slop,berks.verify.dat$Day,power)
plot((berks.dat$Day^power-1)/power , berks.dat$new.cases,main = 'Berks Model1 with Pr
edictions')
prediction.days <- (berks.verify.dat$Day^power-1)/power
prediction <- Yhat(y.int,slop,berks.verify.dat$Day,power)
points(prediction.days, prediction, pch=16,col='red')
```



Hide

```
sprintf('Acutal new case count for %s was %.2f',berks.verify.dat$Date[1],berks.verif
y.dat$new.cases[1])
```

```
## [1] "Acutal new case count for 2020-12-16 was 344.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',berks.verify.dat$Date[1],predictio
n[1])
```

```
## [1] "Predicted new case count for 2020-12-16 was 330.93"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',berks.verify.dat$Date[2],berks.verif
y.dat$new.cases[2])
```

```
## [1] "Acutal new case count for 2020-12-17 was 274.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',berks.verify.dat$Date[2],prediction[2])
```

```
## [1] "Predicted new case count for 2020-12-17 was 337.40"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',berks.verify.dat$Date[3],berks.verify.dat$new.cases[3])
```

```
## [1] "Acutal new case count for 2020-12-18 was 333.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',berks.verify.dat$Date[3],prediction[3])
```

```
## [1] "Predicted new case count for 2020-12-18 was 343.97"
```

Berks County model 2 with non-linear regression (weighted)

Hide

```
residMod1 <- abs(resid(model1))
model2=lm(residMod1~modelDat$Xdat)
fitted=fitted(model2)

weight=(1/(fitted*fitted))

wls=lm(modelDat$new.cases~modelDat$Xdat, weights=weight)
summary(wls)
```

```
##
## Call:
## lm(formula = modelDat$new.cases ~ modelDat$Xdat, weights = weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5880 -0.8010 -0.0474  0.7463  5.0771
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.831e+01  8.972e-01   20.41   <2e-16 ***
## modelDat$Xdat 7.257e-07  3.018e-08   24.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.221 on 195 degrees of freedom
## Multiple R-squared:  0.7478, Adjusted R-squared:  0.7465
## F-statistic: 578.1 on 1 and 195 DF,  p-value: < 2.2e-16
```

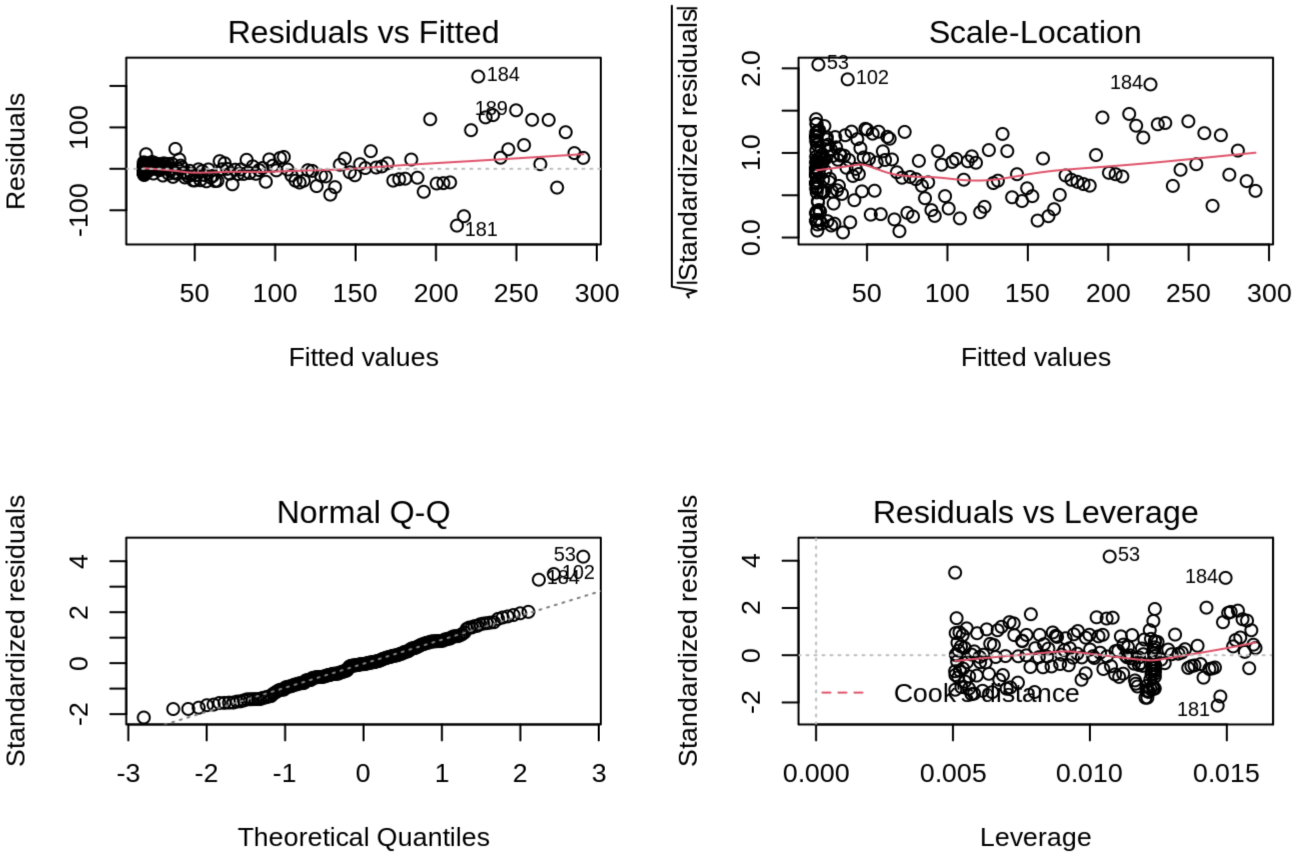
Hide

```
model2.berks <- wls
summary(model2.berks)
```

```
##
## Call:
## lm(formula = modelDat$new.cases ~ modelDat$Xdat, weights = weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5880 -0.8010 -0.0474  0.7463  5.0771
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.831e+01  8.972e-01   20.41   <2e-16 ***
## modelDat$Xdat 7.257e-07  3.018e-08   24.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.221 on 195 degrees of freedom
## Multiple R-squared:  0.7478, Adjusted R-squared:  0.7465
## F-statistic: 578.1 on 1 and 195 DF,  p-value: < 2.2e-16
```

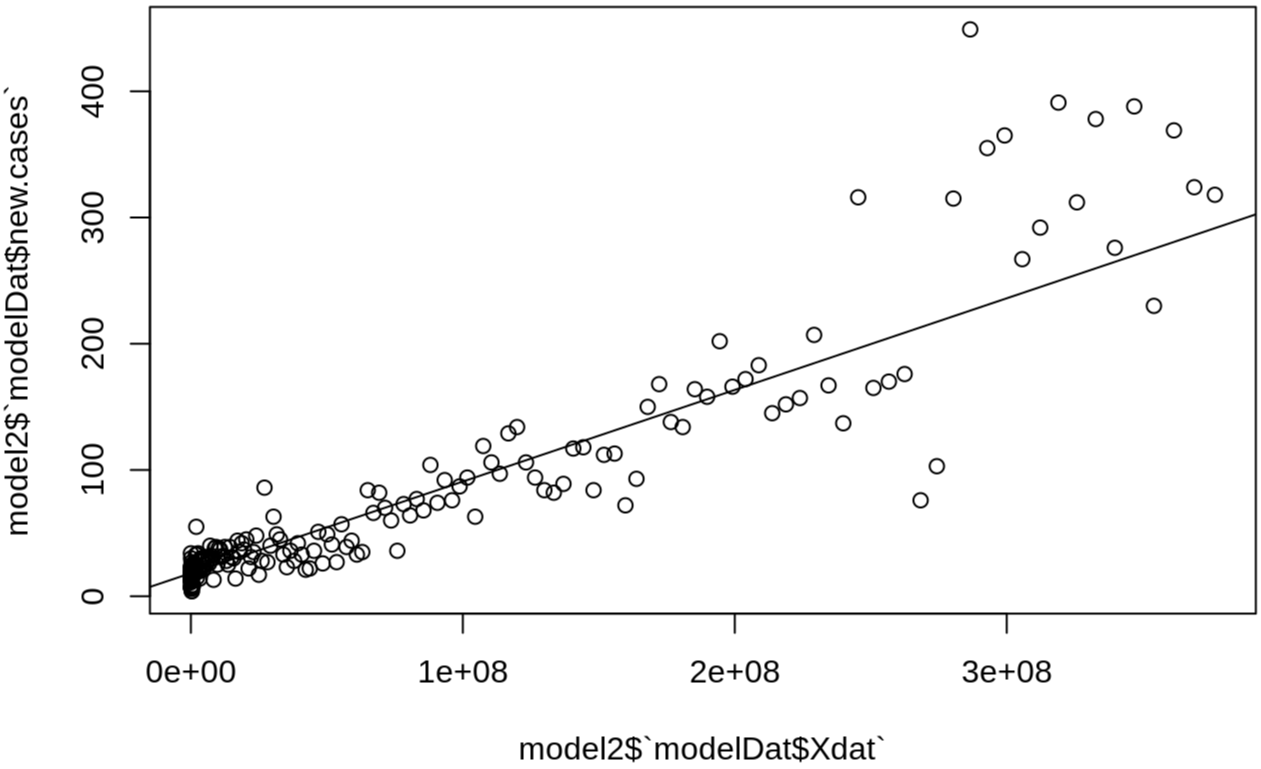
Hide

```
layout(matrix(1:4,2,2))
plot(model2.berks)
```



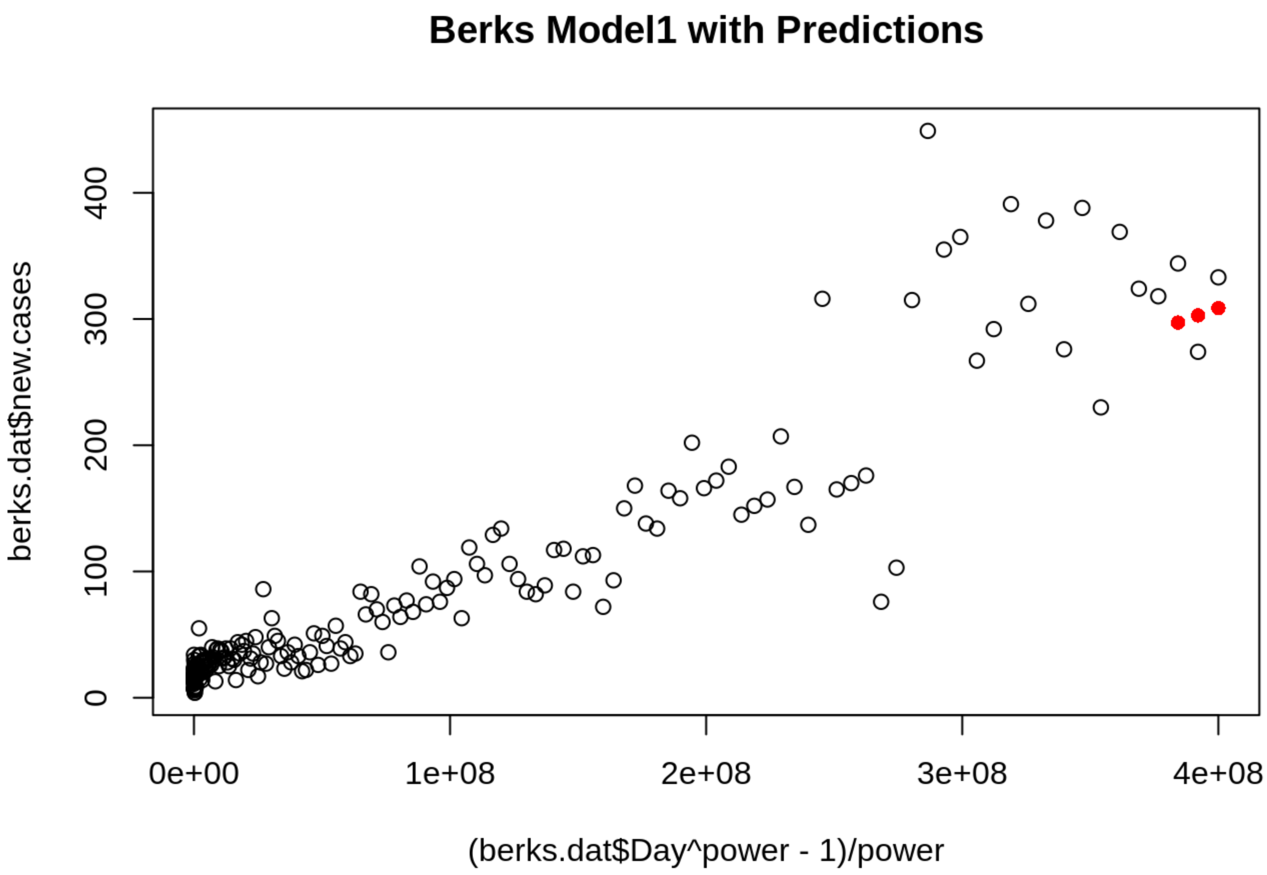
Hide

```
par(mfrow=c(1,1))
model2 <- model2.berks$model
plot(model2$`modelDat$Xdat`,model2$`modelDat$new.cases`)
abline(model2.berks)
```



Hide

```
y.int2 <- model2.berks$coefficients[1]
slop2 <- model2.berks$coefficients[2]
berks.prediction <- Yhat(y.int,slop,berks.verify.dat$Day,power)
plot((berks.dat$Day^power-1)/power , berks.dat$new.cases,main = 'Berks Model1 with Pr
edictions')
prediction.days <- (berks.verify.dat$Day^power-1)/power
prediction <- Yhat(y.int2,slop2,berks.verify.dat$Day,power)
points(prediction.days, prediction, pch=16,col='red')
```



Hide

```
sprintf('Acutal new case count for %s was %.2f',berks.verify.dat$Date[1],berks.verif
y.dat$new.cases[1])
```

```
## [1] "Acutal new case count for 2020-12-16 was 344.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',berks.verify.dat$Date[1],predictio
n[1])
```

```
## [1] "Predicted new case count for 2020-12-16 was 297.16"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',berks.verify.dat$Date[2],berks.verif
y.dat$new.cases[2])
```

```
## [1] "Acutal new case count for 2020-12-17 was 274.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',berks.verify.dat$Date[2],predictio
n[2])
```

```
## [1] "Predicted new case count for 2020-12-17 was 302.84"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',berks.verify.dat$Date[3],berks.verif
y.dat$new.cases[3])
```

```
## [1] "Acutal new case count for 2020-12-18 was 333.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',berks.verify.dat$Date[3],predictio
n[3])
```

```
## [1] "Predicted new case count for 2020-12-18 was 308.60"
```

Philadelphia County model 1 using linear regression and power transformation

Hide

```
# DV  : # of New Cases
# IVs : Day

getMod = function(df){
  p = seq(0,10,0.01)
  Rsquare_p = c(0,0,0,0)
  p.value <- 1
  model = list(NULL)
  for(i in c(2:length(p))){

    modDat <- df %>%  dplyr::mutate(., Xdat = (Day^p[i]-1)/p[i]) %>% drop_na()
    fit <- modDat %>% lm(new.cases ~ Xdat,.)
    sumFit <- summary(fit)
    f <- sumFit$fstatistic
    p.value.fit <- pf(f[1],f[2],f[3],lower.tail=F)
    if(sumFit$adj.r.squared > Rsquare_p[2] & p.value.fit < p.value){
      Rsquare_p <- c(sumFit$r.squared,sumFit$adj.r.squared,p[i],i)
      model <- fit
      p.value <- p.value.fit
      model.sum <- sumFit
    }
  }
  if(is_empty(model)==TRUE){
      print('Search higher power')
      Rsquare_p <- c(sumFit$r.squared,sumFit$adj.r.squared,p[i],i)
      model <- fit
      p.value <- p.value.fit
      model.sum <- sumFit
  }

  return(list(model,Rsquare_p,modDat$Day,p.value,model.sum))
}
philly.model1 <- getMod(philly.model.dat)
```

Explore model 1 for Philadelphia County using linear regression (unweighted)

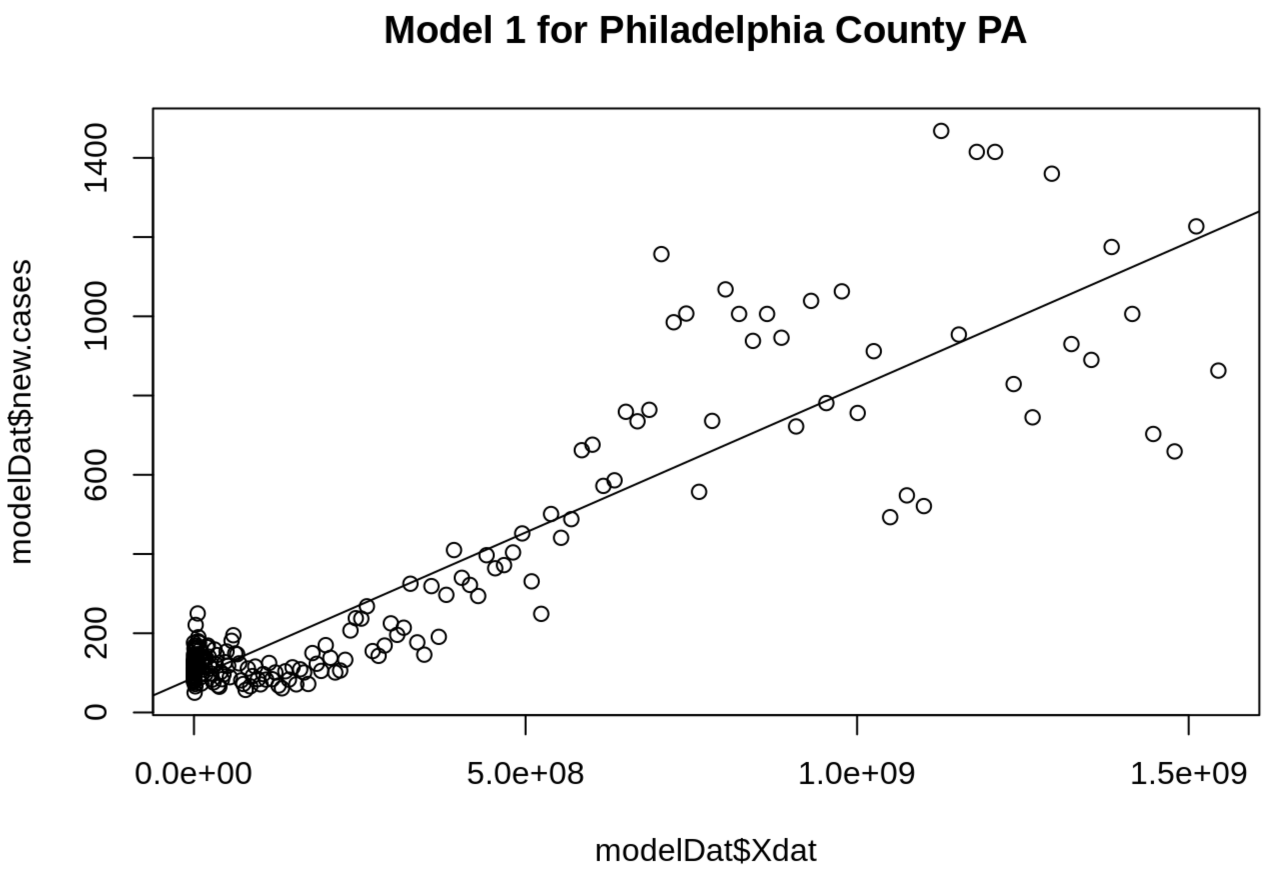
Hide

```
summary(philly.model1[[1]])
```

```
##
## Call:
## lm(formula = new.cases ~ Xdat, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -512.26  -68.97    1.65   40.71  554.55
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.815e+01  1.265e+01    6.97 4.78e-11 ***
## Xdat        7.324e-07  2.501e-08   29.29  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 143.8 on 195 degrees of freedom
## Multiple R-squared:  0.8147, Adjusted R-squared:  0.8138
## F-statistic: 857.6 on 1 and 195 DF,  p-value: < 2.2e-16
```

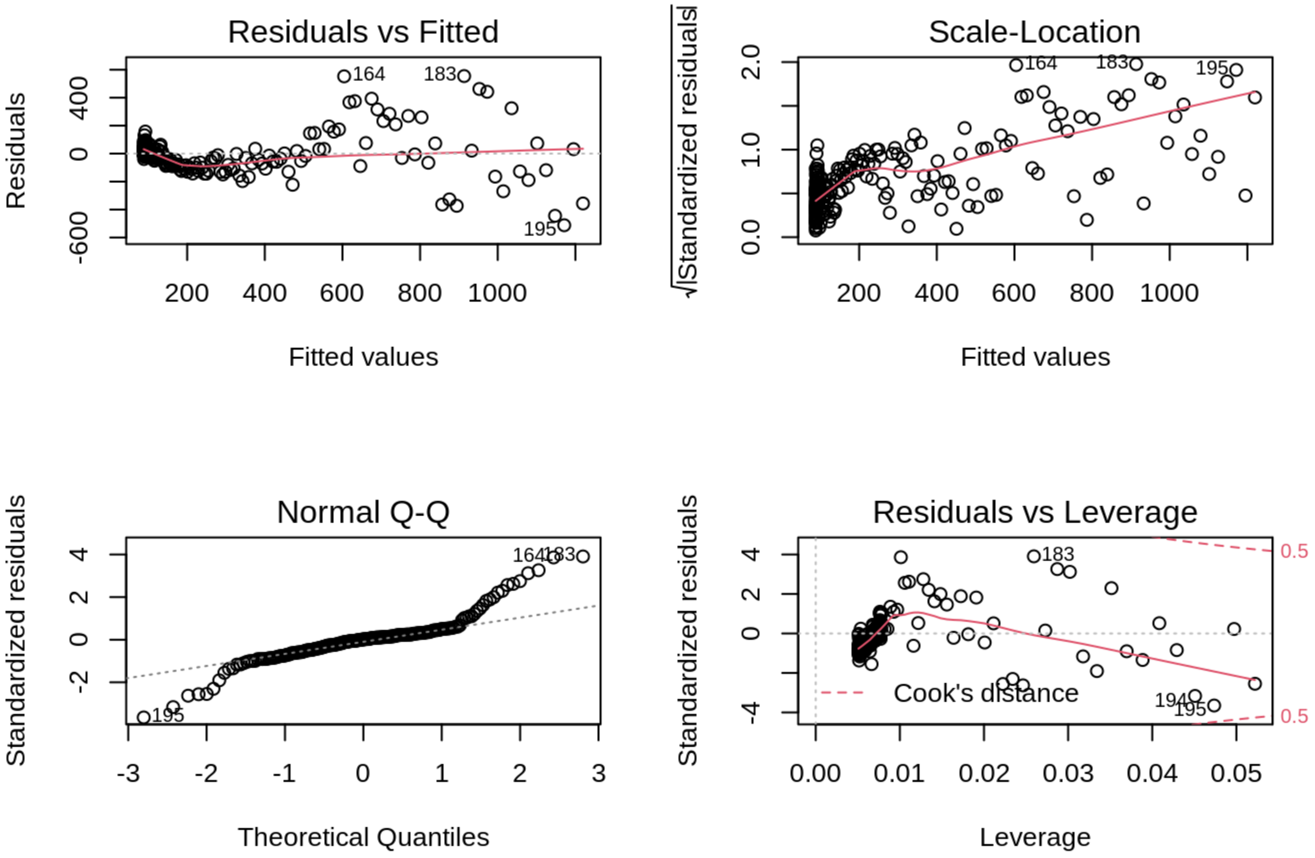
Hide

```
model1 = philly.model1[[1]]
modelDat <- data.frame(model1$model)
power <- philly.model1[[2]][[3]]
plot(modelDat$Xdat,modelDat$new.cases,main = 'Model 1 for Philadelphia County PA')
abline(model1)
```



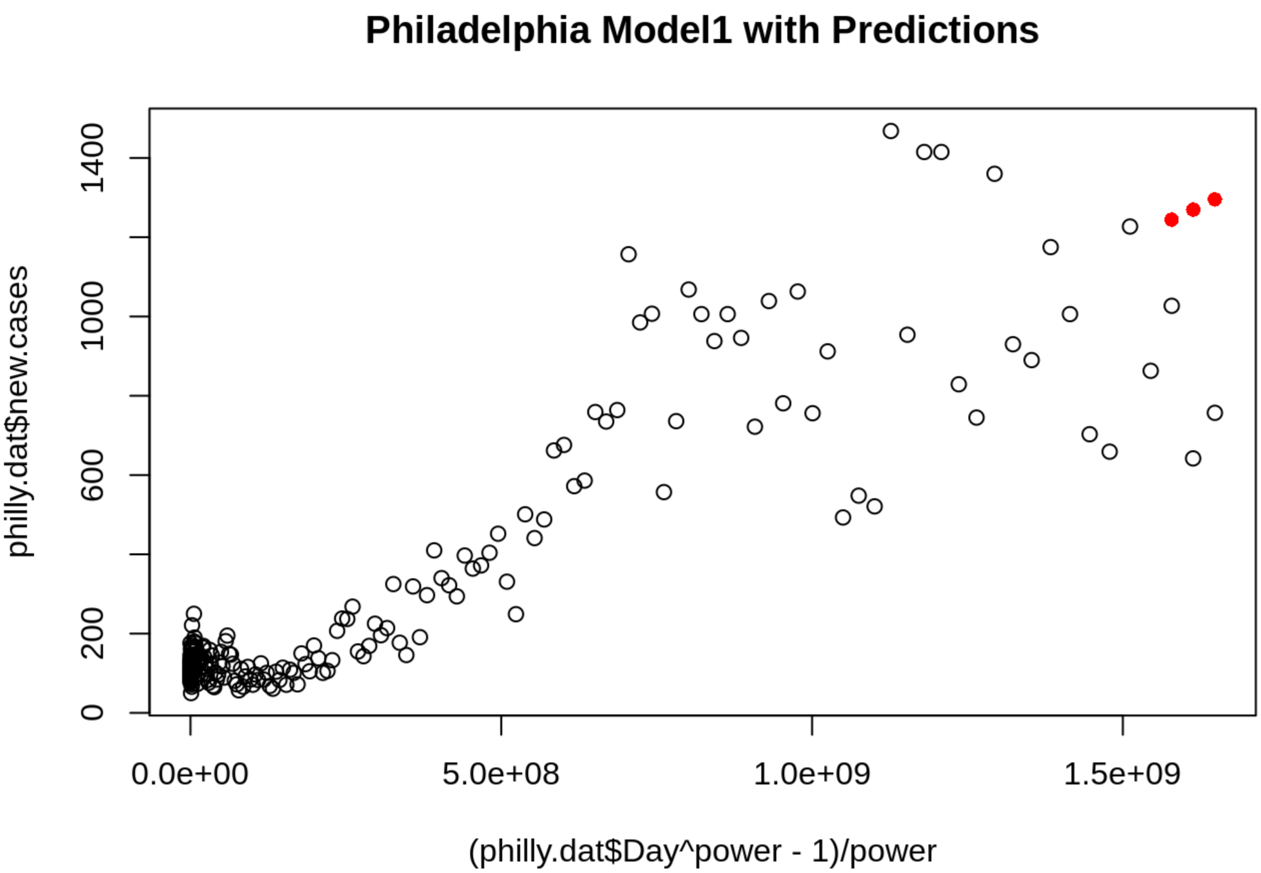
Hide

```
# get summary and plots of fit
layout(matrix(1:4,2,2))
plot(model1)
```



Hide

```
par(mfrow=c(1,1))

# Make predictions
Yhat <- function(B0,B1,day,p){B0+B1*(day^p-1)/p}

y.int <- model1$coefficients[1]
slop <- model1$coefficients[2]
philly.prediction <- Yhat(y.int,slop,philly.verify.dat$Day,power)
plot((philly.dat$Day^power-1)/power , philly.dat$new.cases,main = 'Philadelphia Model 1 with Predictions')
prediction.days <- (philly.verify.dat$Day^power-1)/power
prediction <- Yhat(y.int,slop,philly.verify.dat$Day,power)
points(prediction.days, prediction, pch=16,col='red')
```



Hide

```
sprintf('Acutal new case count for %s was %.2f',philly.verify.dat$Date[1],philly.veri
fy.dat$new.cases[1])
```

```
## [1] "Acutal new case count for 2020-12-16 was 1027.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',philly.verify.dat$Date[1],predicti
on[1])
```

```
## [1] "Predicted new case count for 2020-12-16 was 1244.40"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',philly.verify.dat$Date[2],philly.veri
fy.dat$new.cases[2])
```

```
## [1] "Acutal new case count for 2020-12-17 was 642.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',philly.verify.dat$Date[2],predicti
on[2])
```

```
## [1] "Predicted new case count for 2020-12-17 was 1269.60"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',philly.verify.dat$Date[3],philly.veri
fy.dat$new.cases[3])
```

```
## [1] "Acutal new case count for 2020-12-18 was 757.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',philly.verify.dat$Date[3],predicti
on[3])
```

```
## [1] "Predicted new case count for 2020-12-18 was 1295.22"
```

Create model for entire state of Pennsylvania

Hide

```
# To model the state, parameters need to be combined for a state total for each day
start.date <- '2020-06-01'
end.date <- '2020-12-15'

PA.dat <- rawdata %>%   dplyr::filter(Date >= as.Date(start.date)) %>% drop_na()
Pa.date_as_factor <- data.frame(Day=c(1:length(unique(PA.dat$Date))),Date=unique(PA.d
at$Date))
PA.dat <- PA.dat %>% left_join(.,Pa.date_as_factor,by=c("Date")) %>% group_by(.,Day)
%>% dplyr::mutate(.,Date=unique(Date),Day= Day,new.cases.PA = sum(new.cases), daily.d
eath.count.PA = sum(daily.death.count), patients.hospitalized.PA = sum(patients.hospi
talized), patients.on.ventilators.PA = sum(patients.on.ventilators), population.2018.
PA = sum(population.2018), airborn.isolation.beds.PA = sum(airborn.isolation.beds), a
dult.ICU.beds.PA = sum(adult.ICU.beds)) %>% dplyr::select(.,Date,new.cases.PA, daily.
death.count.PA, patients.hospitalized.PA, Day, patients.on.ventilators.PA, airborn.is
olation.beds.PA, adult.ICU.beds.PA) %>% unique(.) %>% ungroup(.)

PA.model.dat <- PA.dat %>% dplyr::filter(Date <= as.Date(end.date))
PA.verify.dat <- PA.dat %>% dplyr::filter(Date > as.Date(end.date))
```

Hide

```
# Time series model by Day #
# DV = # of new cases per day
# IVS : Day

getMod = function(df){
  p = seq(0,10,.1)
  Rsquare_p = c(0,0,0,0)
  p.value <- 1
  model = list(NULL)
  for(i in c(2:length(p))){

    modDat <- df %>%  dplyr::mutate(., Xdat = (Day^p[i]-1)/p[i]) %>% drop_na()
    #print(head(modDat))
    fit <- lm(new.cases.PA ~ Xdat, data = modDat)
    sumFit <- summary(fit)
    f <- sumFit$fstatistic
    p.value.fit <- pf(f[1],f[2],f[3],lower.tail=F)
    if(sumFit$adj.r.squared > Rsquare_p[2] & p.value.fit < p.value){
      Rsquare_p <- c(sumFit$r.squared,sumFit$adj.r.squared,p[i],i)
      model <- fit
      p.value <- p.value.fit
      model.sum <- sumFit
    }
  }
  if(is_empty(model)==TRUE){
      print('Search higher power')
      Rsquare_p <- c(sumFit$r.squared,sumFit$adj.r.squared,p[i],i)
      model <- fit
      p.value <- p.value.fit
      model.sum <- sumFit
  }

  return(list(model,Rsquare_p,modDat$Day,p.value,model.sum))
}

PA.model1 <- getMod(PA.model.dat)

# PA.model <- lm(new.cases.PA ~ Day + daily.death.count.PA + patients.hospitalized.PA
+ patients.on.ventilators.PA + airborn.isolation.beds.PA + adult.ICU.beds.PA + percen
t.Democrat.PA + percent.Republican.PA, data = PA.model.dat)
```

Explore model 1 for PA

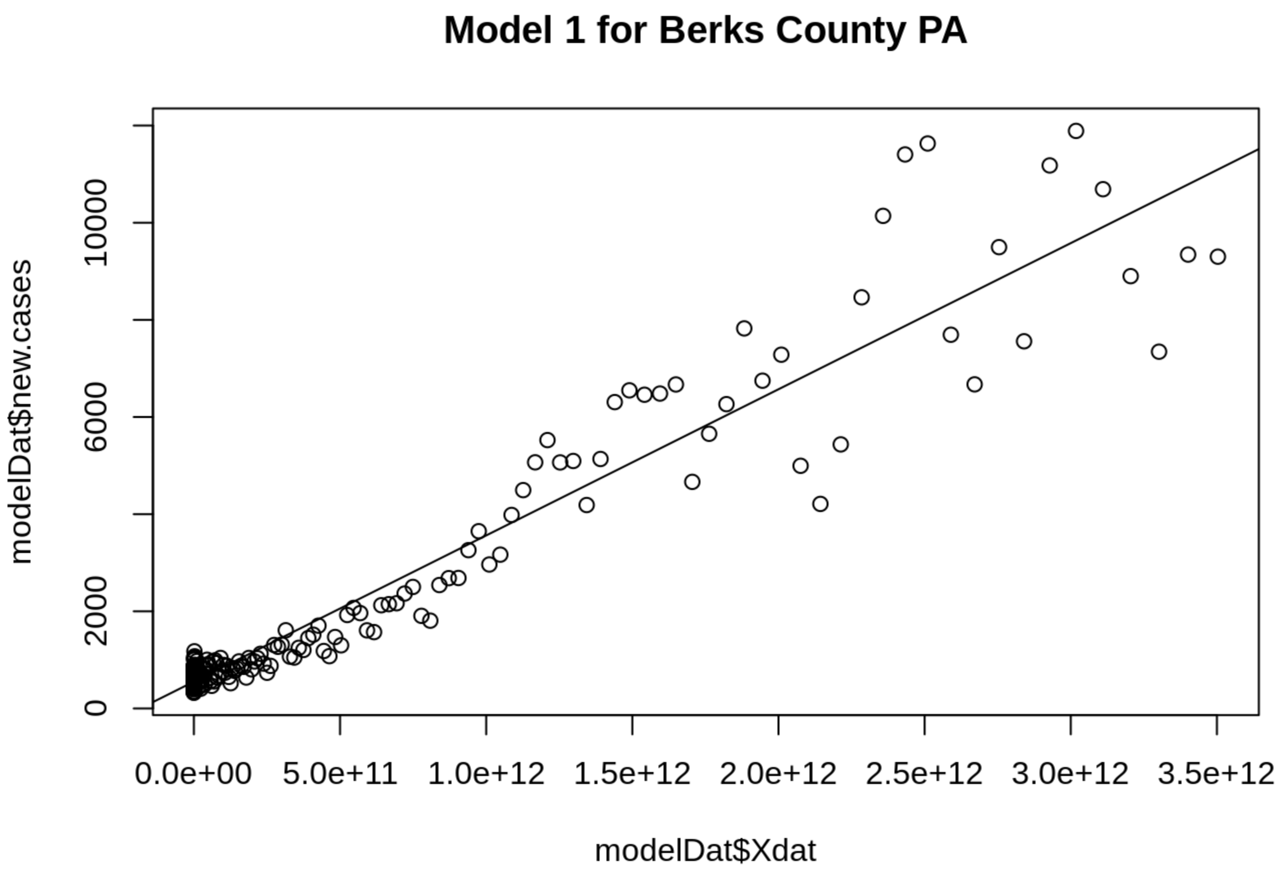
Hide

```
summary(PA.model1[[1]])
```

```
## 
## Call:
## lm(formula = new.cases.PA ~ Xdat, data = modDat)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -3144.6  -210.0   -48.5   213.7  3529.8 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 5.535e+02  6.332e+01   8.741 1.06e-15 ***
## Xdat        3.009e-09  6.315e-11  47.653  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 757.7 on 195 degrees of freedom
## Multiple R-squared:  0.9209, Adjusted R-squared:  0.9205 
## F-statistic:  2271 on 1 and 195 DF,  p-value: < 2.2e-16
```

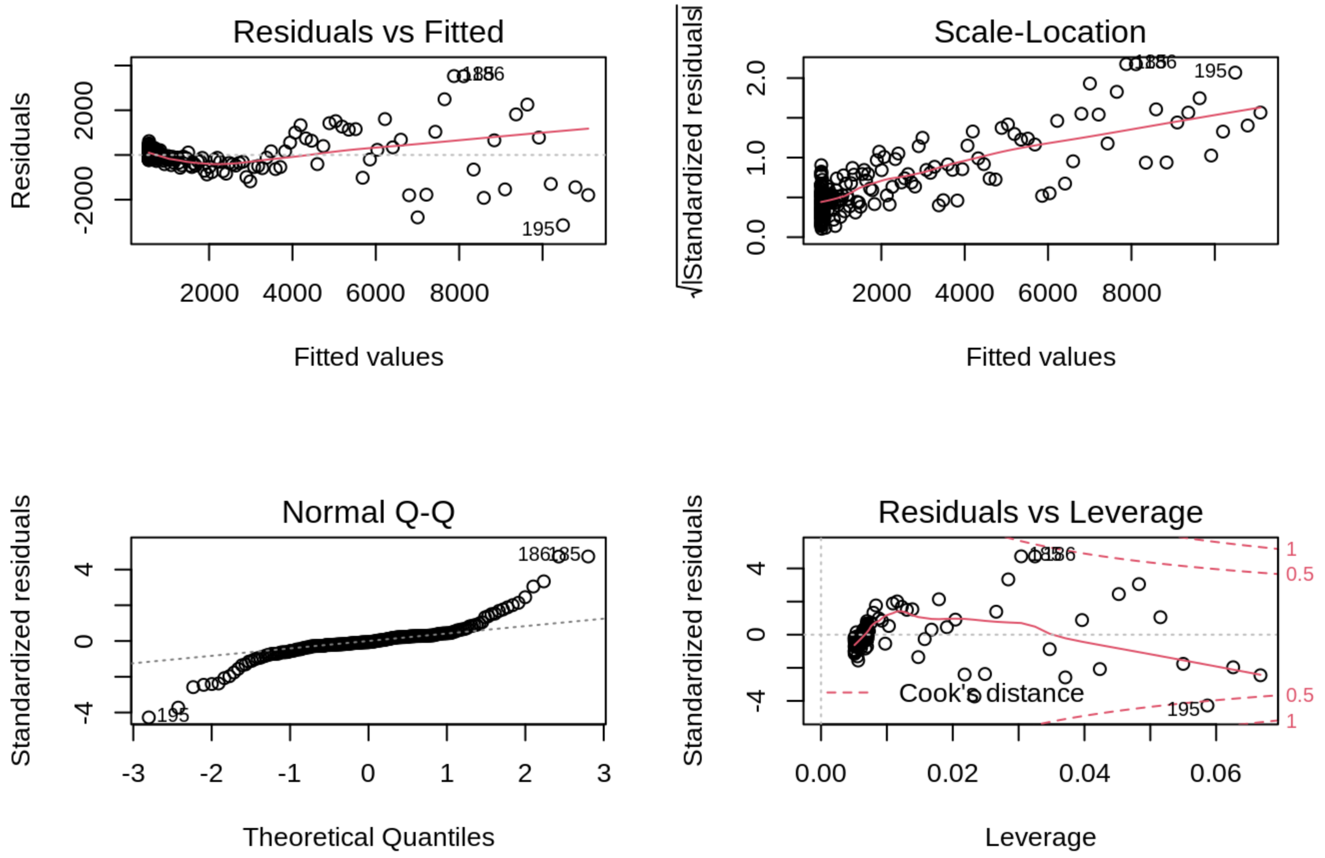
Hide

```
model1 = PA.model1[[1]]
modelDat <- data.frame(model1$model)
power <- PA.model1[[2]][[3]]
plot(modelDat$Xdat,modelDat$new.cases,main = 'Model 1 for Berks County PA')
abline(model1)
```



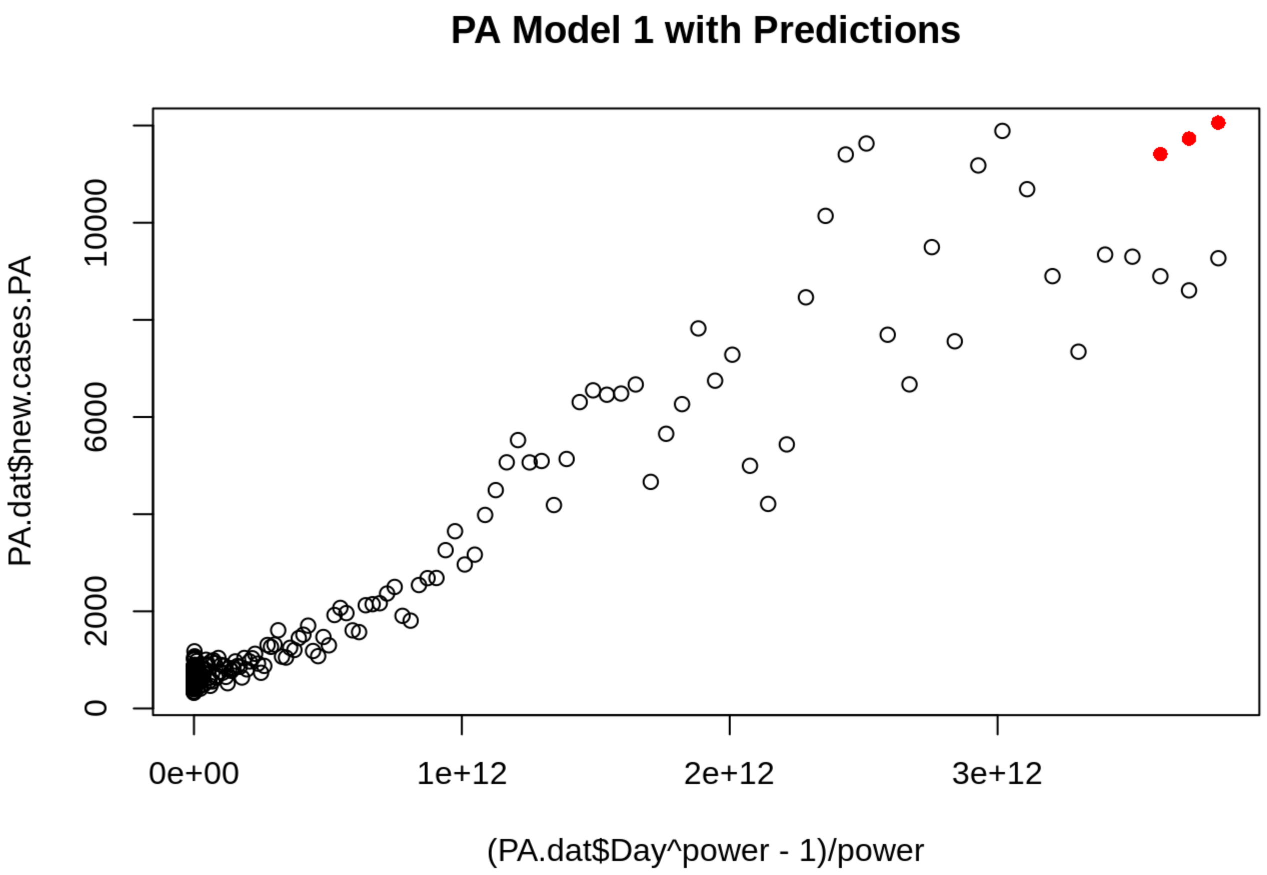
Hide

```
# get summary and plots of fit
layout(matrix(1:4,2,2))
plot(model1)
```



Hide

```
par(mfrow=c(1,1))

# Make predictions
Yhat <- function(B0,B1,day,p){B0+B1*(day^p-1)/p}

y.int <- model1$coefficients[1]
slop <- model1$coefficients[2]

plot((PA.dat$Day^power-1)/power , PA.dat$new.cases.PA,main = 'PA Model 1 with Predictions')
prediction.days <- (PA.verify.dat$Day^power-1)/power
PA.prediction <- Yhat(y.int,slop,PA.verify.dat$Day,power)
points(prediction.days, PA.prediction, pch=16,col='red')
```



Hide

```
sprintf('Acutal new case count for %s was %.2f',PA.verify.dat$Date[1],PA.verify.dat$new.cases.PA[1])
```

```
## [1] "Acutal new case count for 2020-12-16 was 8898.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',PA.verify.dat$Date[1],PA.prediction[1])
```

```
## [1] "Predicted new case count for 2020-12-16 was 11409.54"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',PA.verify.dat$Date[2],PA.verify.dat$new.cases.PA[2])
```

```
## [1] "Acutal new case count for 2020-12-17 was 8607.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',PA.verify.dat$Date[2],PA.prediction[2])
```

```
## [1] "Predicted new case count for 2020-12-17 was 11731.43"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',PA.verify.dat$Date[3],PA.verify.dat$new.cases.PA[3])
```

```
## [1] "Acutal new case count for 2020-12-18 was 9269.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',PA.verify.dat$Date[3],PA.prediction[3])
```

```
## [1] "Predicted new case count for 2020-12-18 was 12061.17"
```

Create weighted model for entire state of Pennsylvania

Hide

```
summary(PA.model1[[1]])
```

```
##
## Call:
## lm(formula = new.cases.PA ~ Xdat, data = modDat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3144.6  -210.0   -48.5   213.7  3529.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.535e+02  6.332e+01   8.741 1.06e-15 ***
## Xdat        3.009e-09  6.315e-11  47.653  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 757.7 on 195 degrees of freedom
## Multiple R-squared:  0.9209, Adjusted R-squared:  0.9205
## F-statistic:  2271 on 1 and 195 DF,  p-value: < 2.2e-16
```

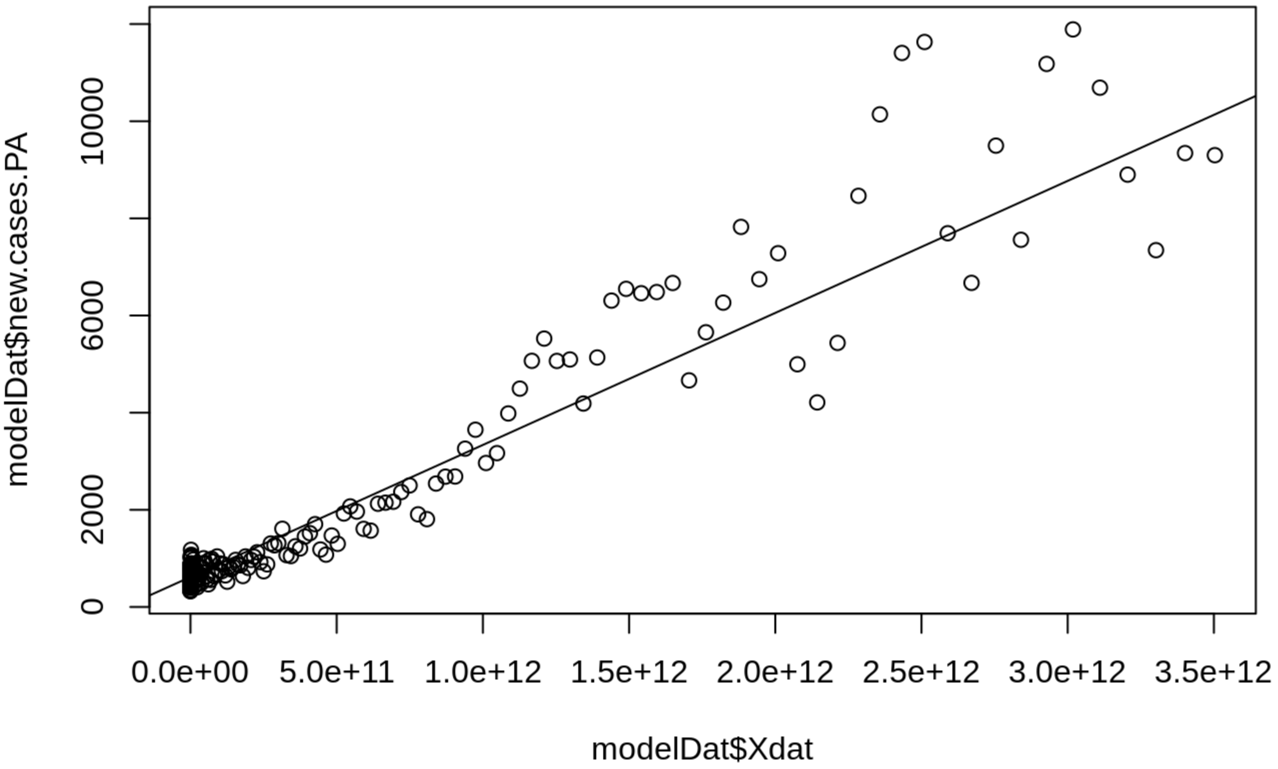
Hide

```
model1 = PA.model1[[1]]
residMod1 <- abs(resid(model1))
model2=lm(residMod1~modelDat$Xdat)
fitted=fitted(model2)

weight=(1/(fitted*fitted))

wls=lm(modelDat$new.cases~modelDat$Xdat, weights=weight)
summary(wls)
```

```
##
## Call:
## lm(formula = modelDat$new.cases ~ modelDat$Xdat, weights = weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9607 -0.8611 -0.2654  0.8833  3.5852
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.182e+02  1.810e+01   34.16   <2e-16 ***
## modelDat$Xdat 2.718e-09  1.011e-10   26.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.186 on 195 degrees of freedom
## Multiple R-squared:  0.7876, Adjusted R-squared:  0.7865
## F-statistic: 722.9 on 1 and 195 DF,  p-value: < 2.2e-16
```

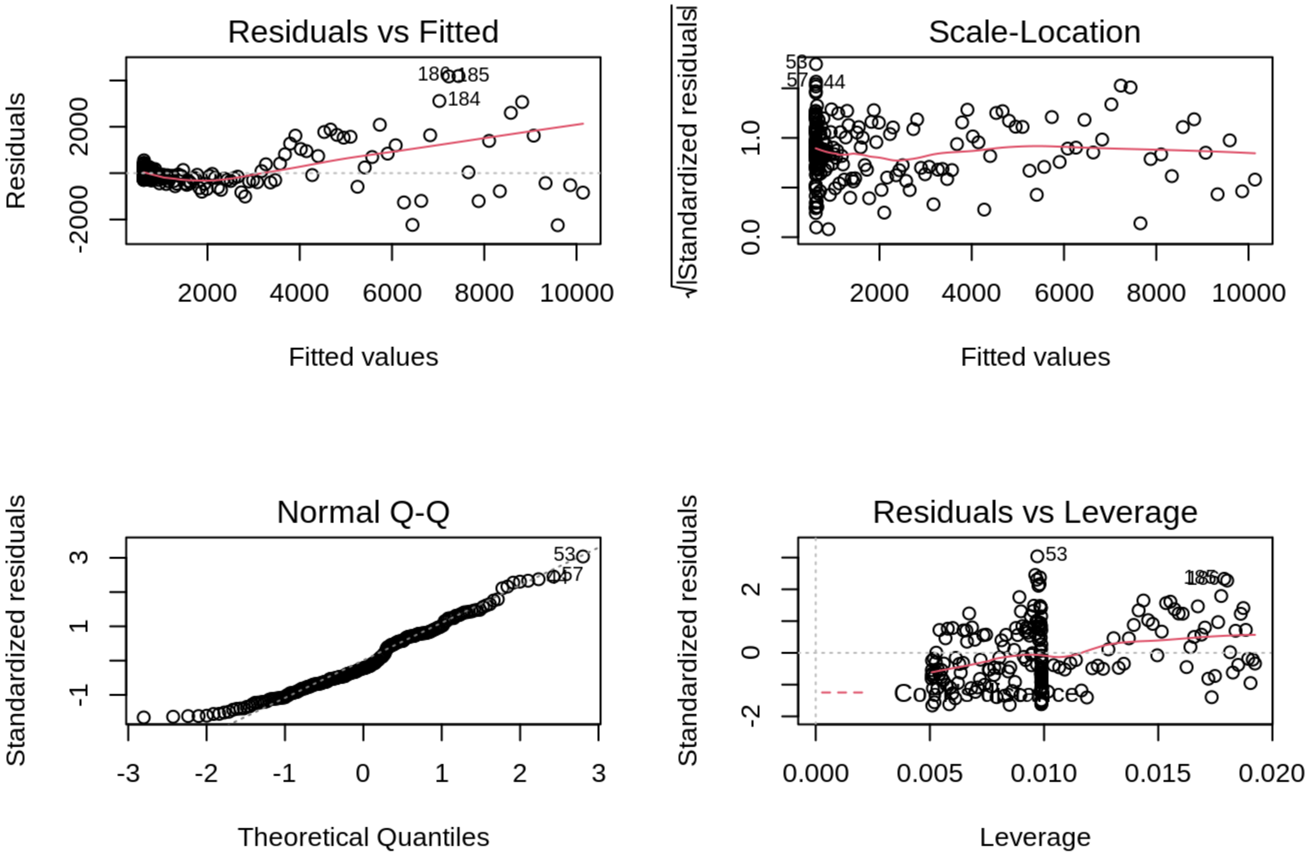
Hide

```
#plot(wls)
plot(modelDat$Xdat,modelDat$new.cases.PA)
abline(wls)
```



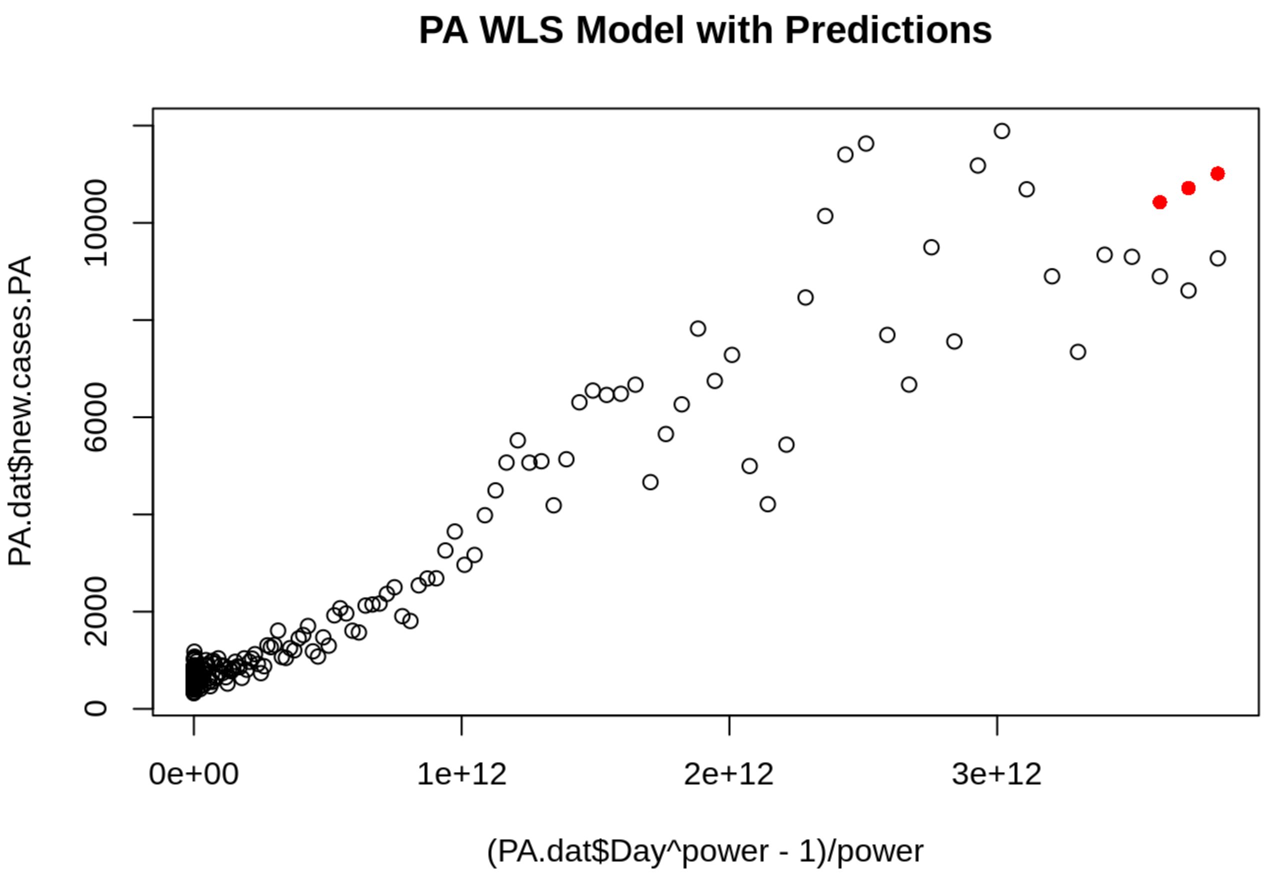
Hide

```
power <- PA.model1[[2]][[3]]
# get summary and plots of fit
layout(matrix(1:4,2,2))
plot(wls)
```



Hide

```
par(mfrow=c(1,1))
#
# # Make predictions
# Yhat <- function(B0,B1,day,p){B0+B1*(day^p-1)/p}
#
y.int <- wls$coefficients[1]
slop <- wls$coefficients[2]
#
plot((PA.dat$Day^power-1)/power , PA.dat$new.cases.PA,main = 'PA WLS Model with Predi
ctions')
prediction.days <- (PA.verify.dat$Day^power-1)/power
PA.prediction.wls <- Yhat(y.int,slop,PA.verify.dat$Day,power)
points(prediction.days, PA.prediction.wls, pch=16,col='red')
```



Hide

```
sprintf('Acutal new case count for %s was %.2f',PA.verify.dat$Date[1],PA.verify.dat$new.cases.PA[1])
```

```
## [1] "Acutal new case count for 2020-12-16 was 8898.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',PA.verify.dat$Date[1],PA.prediction.wls[1])
```

```
## [1] "Predicted new case count for 2020-12-16 was 10423.85"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',PA.verify.dat$Date[2],PA.verify.dat$new.cases.PA[2])
```

```
## [1] "Acutal new case count for 2020-12-17 was 8607.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',PA.verify.dat$Date[2],PA.prediction.wls[2])
```

```
## [1] "Predicted new case count for 2020-12-17 was 10714.59"
```

Hide

```
sprintf('Acutal new case count for %s was %.2f',PA.verify.dat$Date[3],PA.verify.dat$new.cases.PA[3])
```

```
## [1] "Acutal new case count for 2020-12-18 was 9269.00"
```

Hide

```
sprintf('Predicted new case count for %s was %.2f',PA.verify.dat$Date[3],PA.prediction.wls[3])
```

```
## [1] "Predicted new case count for 2020-12-18 was 11012.43"
```