Deal or No Deal

Loading packages and data

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(plm)
```

```
##
## Attaching package: 'plm'

## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 3.6.2

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.6.2

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(mgcv)
```

```
## Loading required package: nlme

##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse

## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.
```

```r
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
df <- read_excel("/Users/jake/Desktop/Applied Statistics II/Deal_Or_No_Deal.xls", sheet = "US")
df2 <- read_excel("/Users/jake/Desktop/Applied Statistics II/Deal_Or_No_Deal.xls", sheet = "USX")
```

CLeaning df

```r
case_values <- colnames(df[12:37]) %>% as.integer()
df$cases_remaining <- rowSums(df[, 12:37])

for (i in 1:nrow(df)){
  df[i, 12:37] <- df[i, 12:37] * case_values
}

df$exp_value <- rowSums(df[, 12:37]) / df$cases_remaining

df$max <- NA
for (i in 1:nrow(df)){
  df[i, "max"] <- df[i, 12:37] %>% max()
}

df$rms <- sqrt(rowSums((df[, 12:37])^2) / df$cases_remaining)

df <- df[,c(3:11, 38:41)]
colnames(df)[5:9] <- c("stop_round","amount_won", "round", "deal_nodeal", "bank_offer")
```

CLeaning df2

```r
df2 <- df2[-which(is.na(df2$`ID Number`)), ]
df2[which(df2$Name == "Cindy"), "Name"] <- "Cindy2"

case_values2 <- colnames(df2[12:37]) %>% as.integer()
df2$cases_remaining <- rowSums(df2[, 12:37])

for (i in 1:nrow(df2)){
  df2[i, 12:37] <- df2[i, 12:37] * case_values2
}

df2$exp_value <- rowSums(df2[, 12:37]) / df2$cases_remaining

df2$max <- NA
for (i in 1:nrow(df2)){
  df2[i, "max"] <- df2[i, 12:37] %>% max()
}

df2$rms <- sqrt(rowSums((df2[, 12:37])^2) / df2$cases_remaining)

df2 <- df2[,c(3:11, 38:41)]
colnames(df2)[5:9] <- c("stop_round","amount_won", "round", "deal_nodeal", "bank_offer")
```

Joining df and df2

```
df <- full_join(df, df2, by = colnames(df))

df$exp_sq <- (df$exp_value)^2
df$round <- df$round %>% as.factor()
```

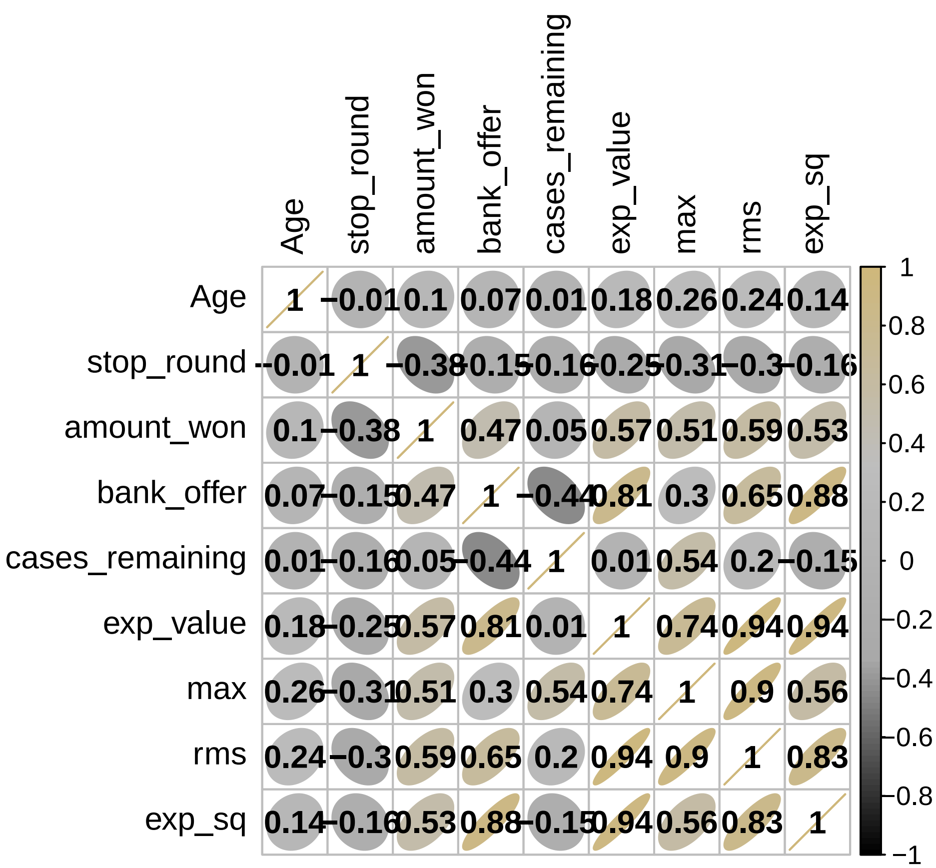
Splitting into training and tetsing

```
set.seed(23)
ppl <- df$Name %>% unique()
ppl_count <- ppl %>% length()
index2 <- sort(sample(ppl_count, ppl_count*.7))
train_names <- ppl[index2]

train2 <- df[which(df$Name %in% train_names), ]
test2 <- df[-which(df$Name %in% train_names), ]
```

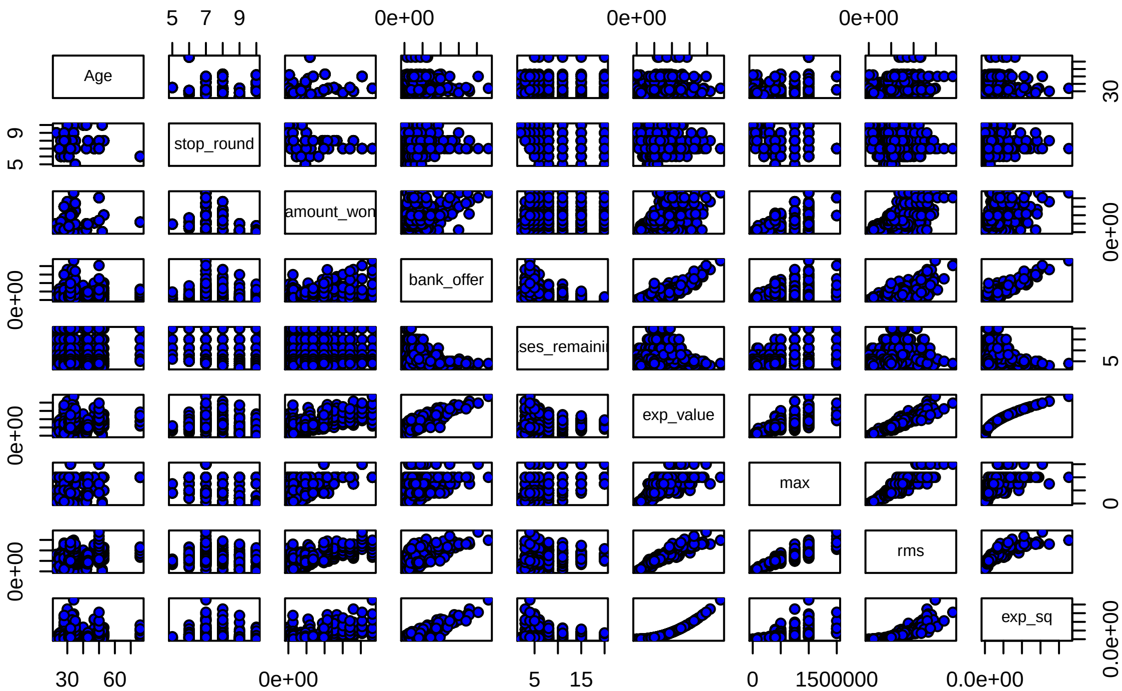
EDA

```
col4 = colorRampPalette(c("black", "darkgrey", "grey", "#CFB87C"))
corrplot(cor(train2[,c(4:6,9:14)]), method = "ellipse", col = col4(100), addCoef.col = "black", tl.col =
```

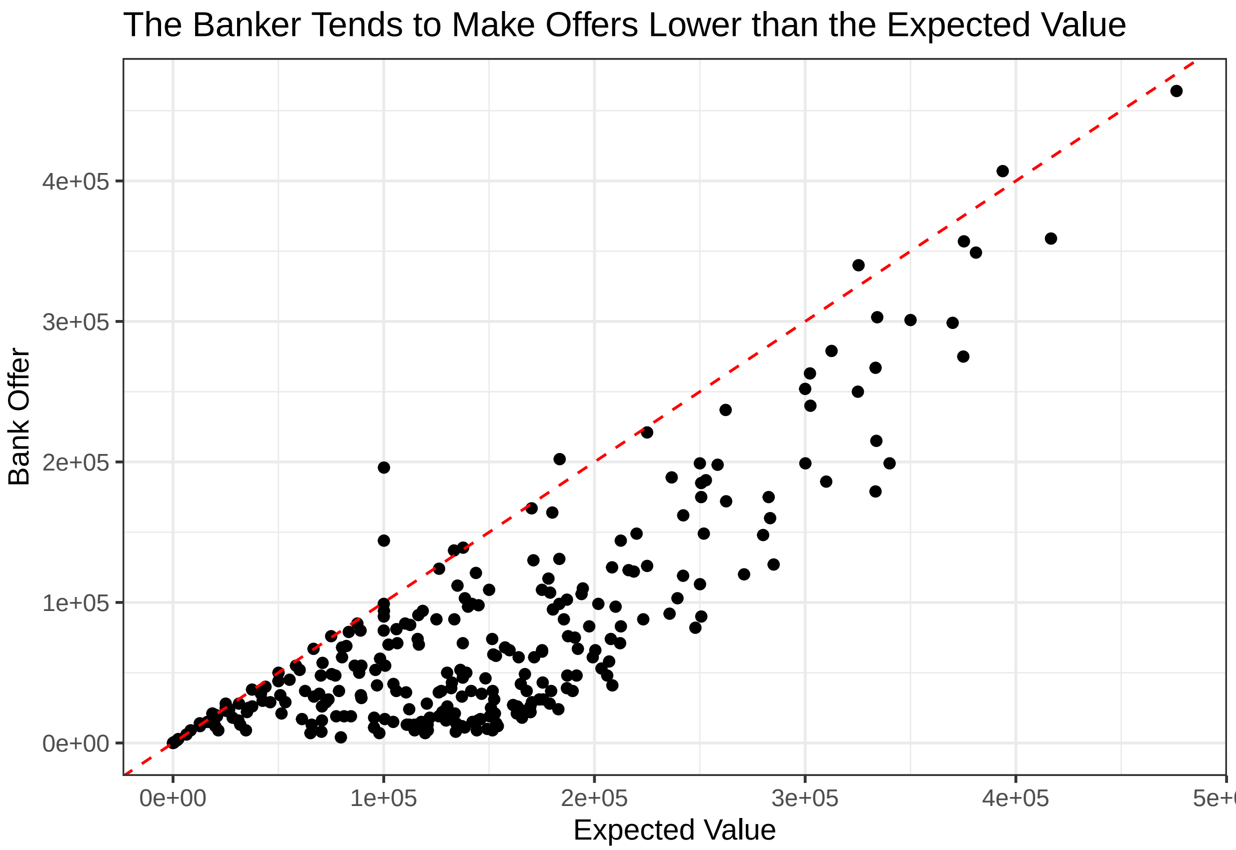



```
pairs(train2[,c(4:6,9:14)], main = "Data", pch = 21, bg = c("blue"))
```

Data

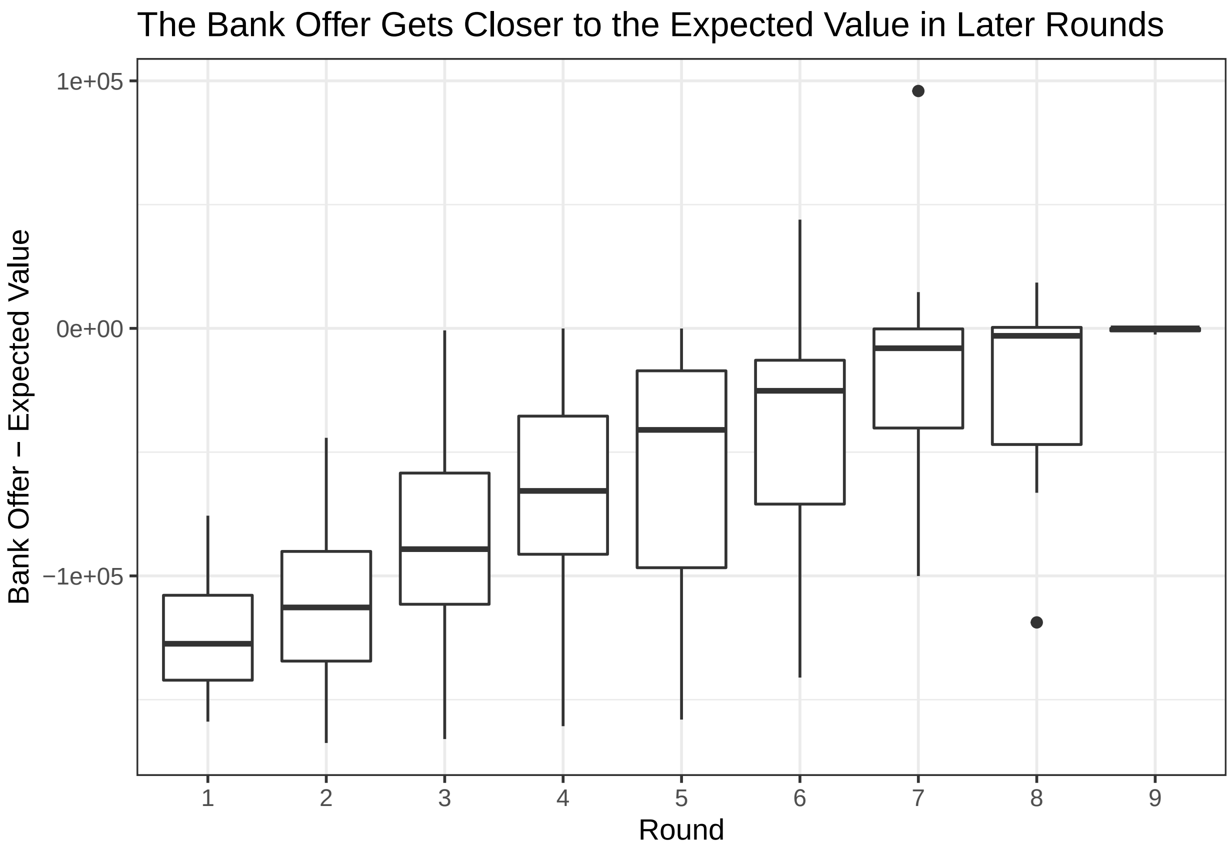



```r
ggplot(train2, aes(exp_value, bank_offer)) +
  geom_point() +
  xlab("Expected Value") +
  ylab("Bank Offer") +
  ggtitle("The Banker Tends to Make Offers Lower than the Expected Value") +
  geom_abline(slope = 1, intercept = 0, lty = 2, col = "red") +
  theme_bw()
```



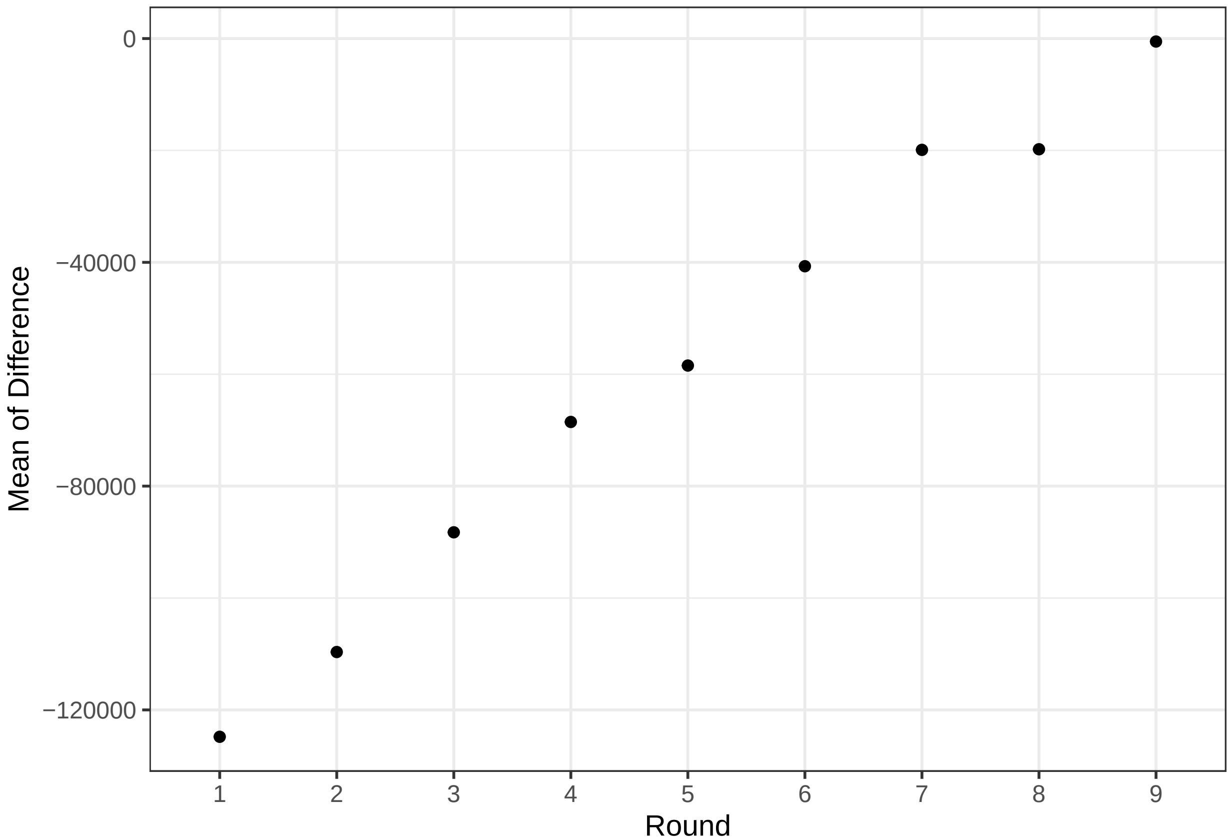

```r
train2$diff <- train2$bank_offer - train2$exp_value

ggplot(train2, aes(x = round, y = diff)) +
  geom_boxplot() +
  xlab("Round") +
  ylab("Bank Offer - Expected Value") +
  ggtitle("The Bank Offer Gets Closer to the Expected Value in Later Rounds") +
  theme_bw()
```

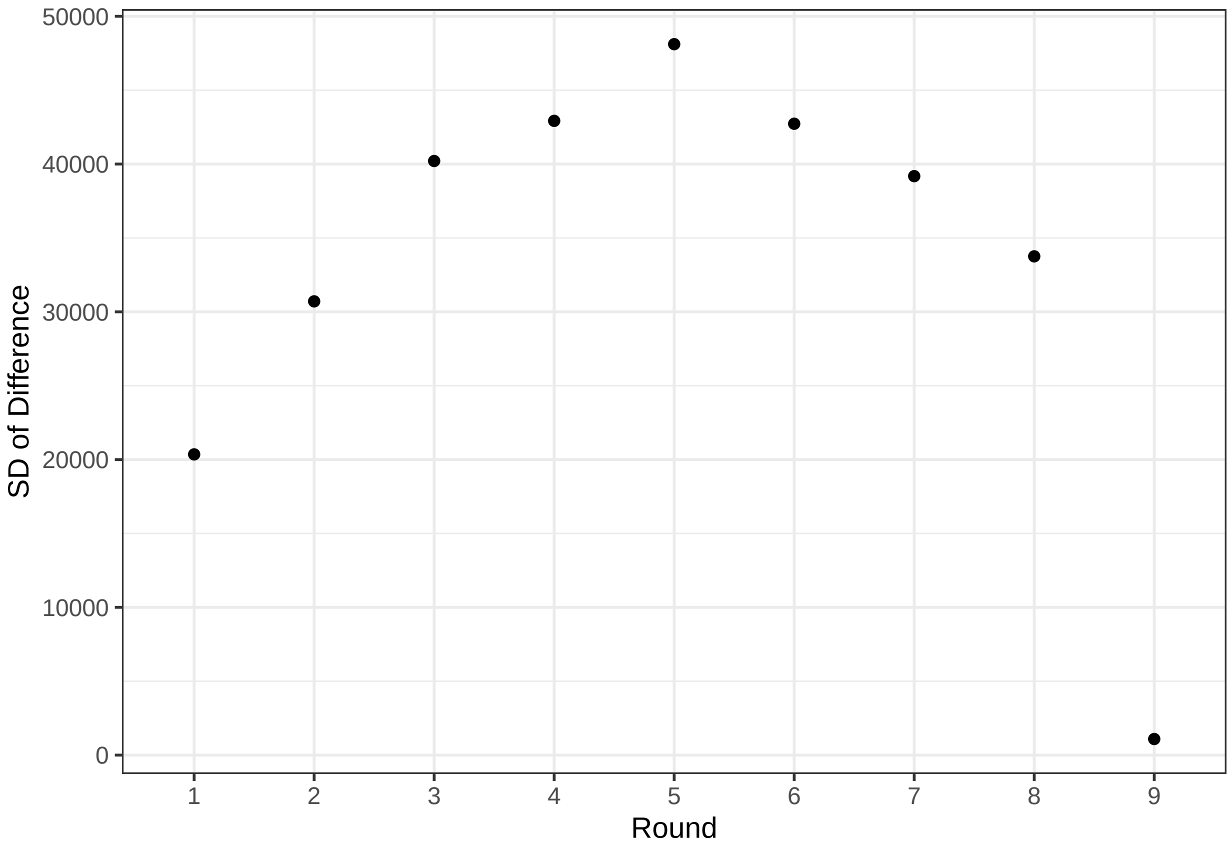



```
train2$diff <- train2$bank_offer - train2$exp_value
agg <- aggregate(train2$diff, list(train2$round), mean)
agg2 <- aggregate(train2$diff, list(train2$round), sd)

ggplot(agg, aes(Group.1, x)) +
  geom_point() +
  xlab("Round") +
  ylab("Mean of Difference") +
  theme_bw()
```




```
ggplot(agg2, aes(Group.1, x)) +
  geom_point() +
  xlab("Round") +
  ylab("SD of Difference") +
  theme_bw()
```



Testing linear models

```
lmod <- lm(bank_offer ~ I(exp_value^2), train2)
summary(lmod)
```

```
##
## Call:
## lm(formula = bank_offer ~ I(exp_value^2), data = train2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -67687 -28877  -2557  23049 159428
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.575e+04  2.930e+03   5.376 1.61e-07 ***
## I(exp_value^2) 2.078e-06  6.676e-08  31.122  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37470 on 279 degrees of freedom
## Multiple R-squared:  0.7764, Adjusted R-squared:  0.7756
## F-statistic: 968.6 on 1 and 279 DF,  p-value: < 2.2e-16
```

```
lmod2 <- lm(bank_offer ~ I(exp_value^2) + round, train2)
summary(lmod2)
```

```
##
## Call:
```

```
## lm(formula = bank_offer ~ I(exp_value^2) + round, data = train2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -69330 -14095   -956  11541 117030
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.310e+04  4.596e+03  -5.027 9.08e-07 ***
## I(exp_value^2)  1.905e-06  5.064e-08  37.610  < 2e-16 ***
## round2          1.376e+04  6.335e+03   2.171   0.0308 *
## round3          3.166e+04  6.336e+03   4.997 1.05e-06 ***
## round4          4.564e+04  6.337e+03   7.202 5.87e-12 ***
## round5          5.406e+04  6.360e+03   8.499 1.29e-15 ***
## round6          7.116e+04  6.437e+03  11.054  < 2e-16 ***
## round7          8.299e+04  6.768e+03  12.263  < 2e-16 ***
## round8          6.761e+04  7.584e+03   8.915  < 2e-16 ***
## round9          4.919e+04  1.017e+04   4.838 2.21e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27240 on 271 degrees of freedom
## Multiple R-squared:  0.8852, Adjusted R-squared:  0.8814
## F-statistic: 232.1 on 9 and 271 DF,  p-value: < 2.2e-16
```

```r
lmod3 <- lm(bank_offer ~ I(exp_value^2) + round + Gender, train2)
summary(lmod3)
```

```
##
## Call:
## lm(formula = bank_offer ~ I(exp_value^2) + round + Gender, data = train2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -72308 -13977   -391  10950 119591
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.580e+04  4.797e+03  -5.378 1.63e-07 ***
## I(exp_value^2)  1.899e-06  5.049e-08  37.620  < 2e-16 ***
## round2          1.377e+04  6.306e+03   2.183   0.0299 *
## round3          3.167e+04  6.307e+03   5.022 9.31e-07 ***
## round4          4.566e+04  6.308e+03   7.238 4.73e-12 ***
## round5          5.412e+04  6.331e+03   8.548 9.43e-16 ***
## round6          7.134e+04  6.408e+03  11.132  < 2e-16 ***
## round7          8.318e+04  6.737e+03  12.346  < 2e-16 ***
## round8          6.803e+04  7.553e+03   9.008  < 2e-16 ***
## round9          4.986e+04  1.013e+04   4.923 1.48e-06 ***
## GenderM         6.101e+03  3.263e+03   1.870   0.0625 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27120 on 270 degrees of freedom
## Multiple R-squared:  0.8866, Adjusted R-squared:  0.8824
## F-statistic: 211.2 on 10 and 270 DF,  p-value: < 2.2e-16
```

```r
lmod4 <- lm(bank_offer ~ I(exp_value^2) + round + max, train2)
summary(lmod4)
```

```
##
## Call:
## lm(formula = bank_offer ~ I(exp_value^2) + round + max, data = train2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -68557 -13846  -1139  11248 116051
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.708e+04  8.422e+03  -3.215  0.00146 **
## I(exp_value^2)  1.868e-06  8.259e-08  22.615  < 2e-16 ***
## round2          1.428e+04  6.410e+03   2.227  0.02675 *
## round3          3.274e+04  6.628e+03   4.940 1.37e-06 ***
## round4          4.738e+04  7.059e+03   6.713 1.12e-10 ***
## round5          5.643e+04  7.637e+03   7.390 1.85e-12 ***
## round6          7.382e+04  7.993e+03   9.236  < 2e-16 ***
## round7          8.640e+04  9.079e+03   9.516  < 2e-16 ***
## round8          7.126e+04  9.974e+03   7.144 8.45e-12 ***
## round9          5.300e+04  1.223e+04   4.335 2.06e-05 ***
## max             4.791e-03  8.497e-03   0.564  0.57336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27280 on 270 degrees of freedom
## Multiple R-squared:  0.8853, Adjusted R-squared:  0.8811
## F-statistic: 208.4 on 10 and 270 DF,  p-value: < 2.2e-16
```

```r
anova(lmod2, lmod3)
```

```
## Analysis of Variance Table
##
## Model 1: bank_offer ~ I(exp_value^2) + round
## Model 2: bank_offer ~ I(exp_value^2) + round + Gender
##   Res.Df        RSS Df  Sum of Sq      F  Pr(>F)
## 1    271 2.0116e+11
## 2    270 1.9859e+11  1 2572303823 3.4973 0.06255 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
fixed <- plm(bank_offer ~ I(exp_value^2), data=train2, index=c("Name"), model="within")
summary(fixed)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = bank_offer ~ I(exp_value^2), data = train2, model = "within",
##     index = c("Name"))
##
## Unbalanced Panel: n = 35, T = 5-16, N = 281
##
## Residuals:
```

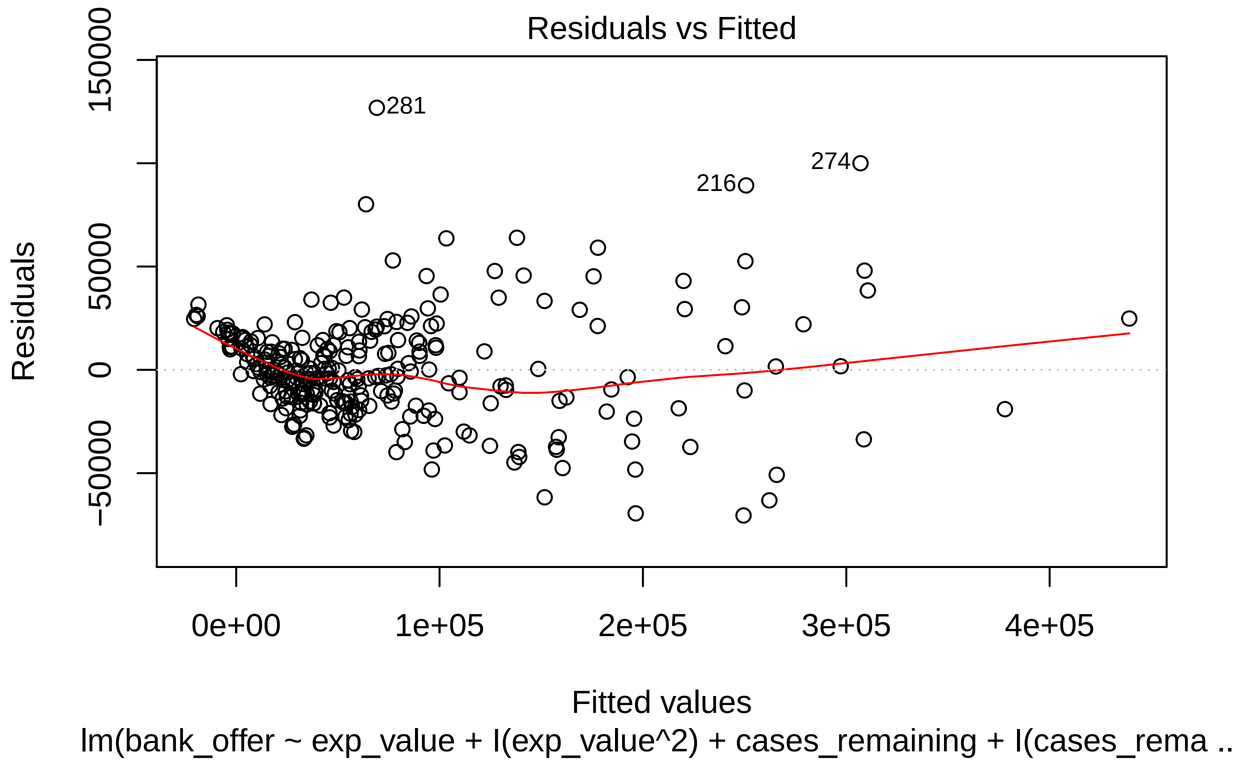
```
##    Min. 1st Qu.  Median 3rd Qu.    Max.
##  -91210  -27507    1791   20513  124967
##
## Coefficients:
##                   Estimate Std. Error t-value  Pr(>|t|)
## I(exp_value^2) 2.2035e-06 8.5725e-08  25.704 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    1.2114e+12
## Residual Sum of Squares: 3.2769e+11
## R-Squared:      0.7295
## Adj. R-Squared: 0.69085
## F-statistic: 660.719 on 1 and 245 DF, p-value: < 2.22e-16
```

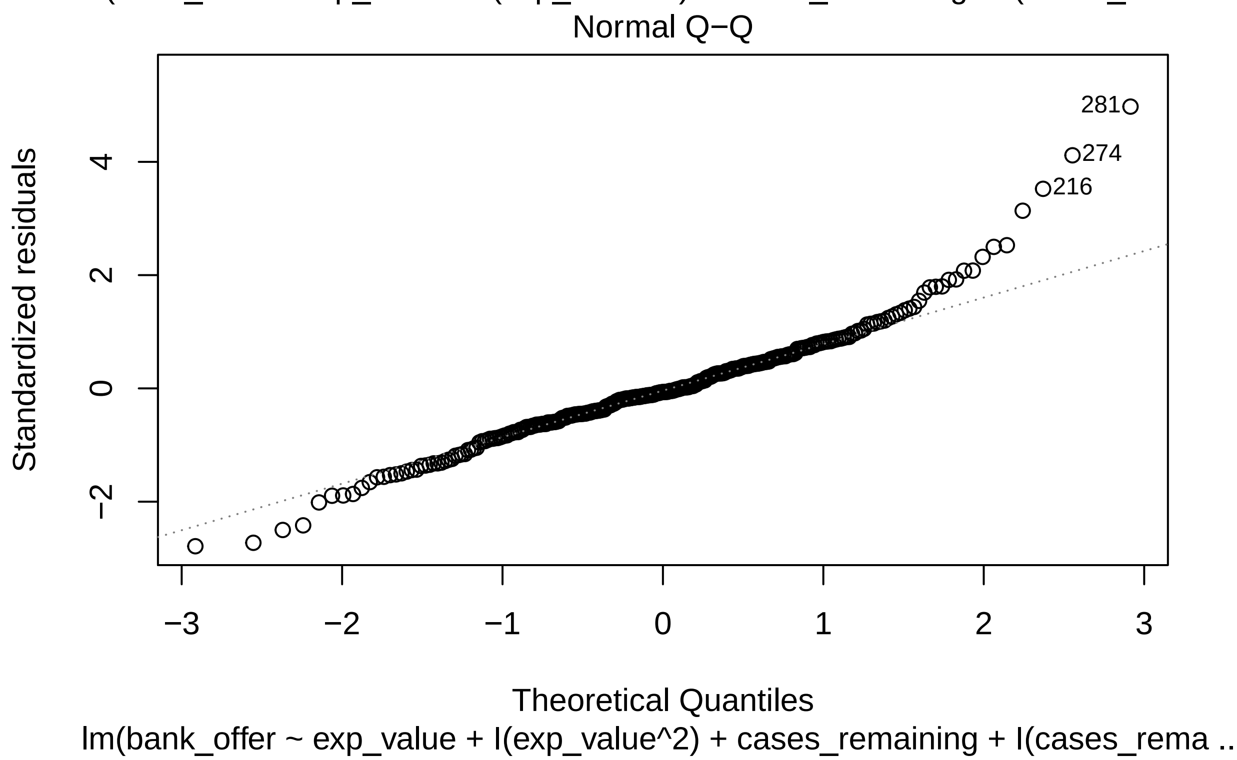
Online Model used in paper

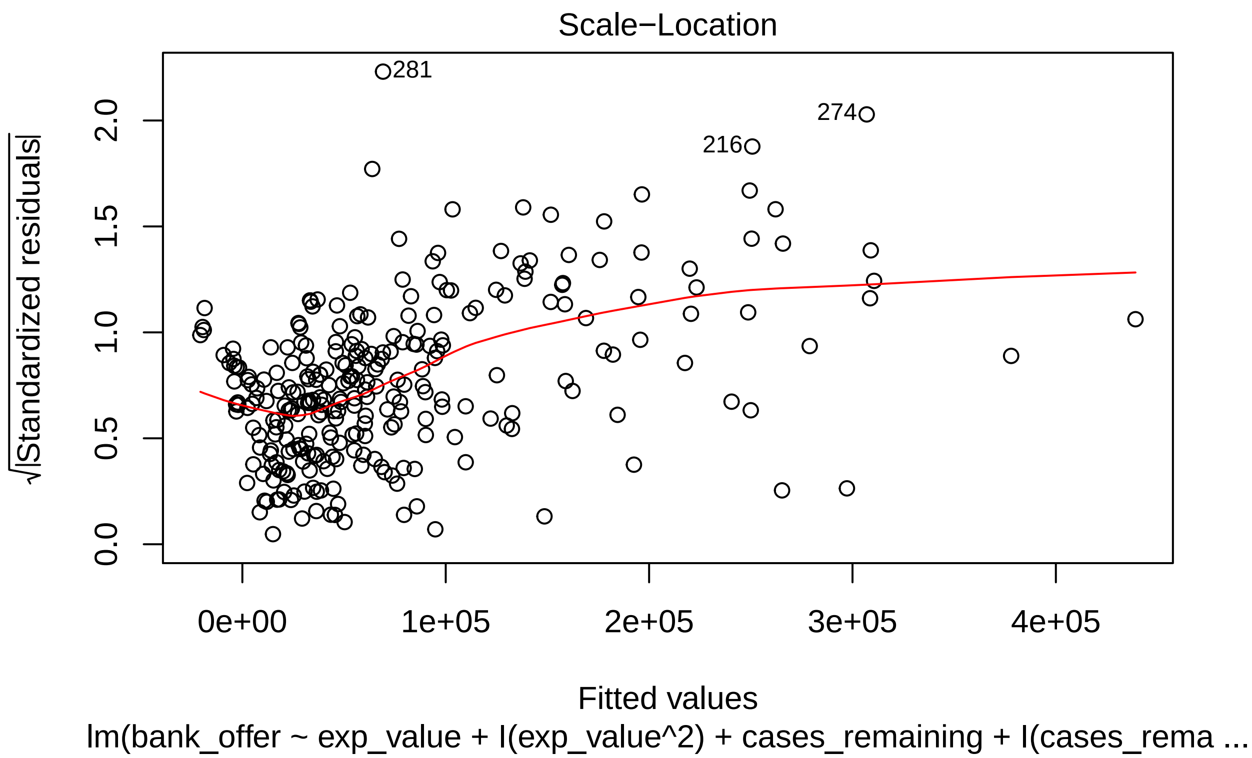
```
online1 <- lm(bank_offer ~ exp_value + I(exp_value^2) + cases_remaining + I(cases_remaining^2) + max, t
summary(online1)
```

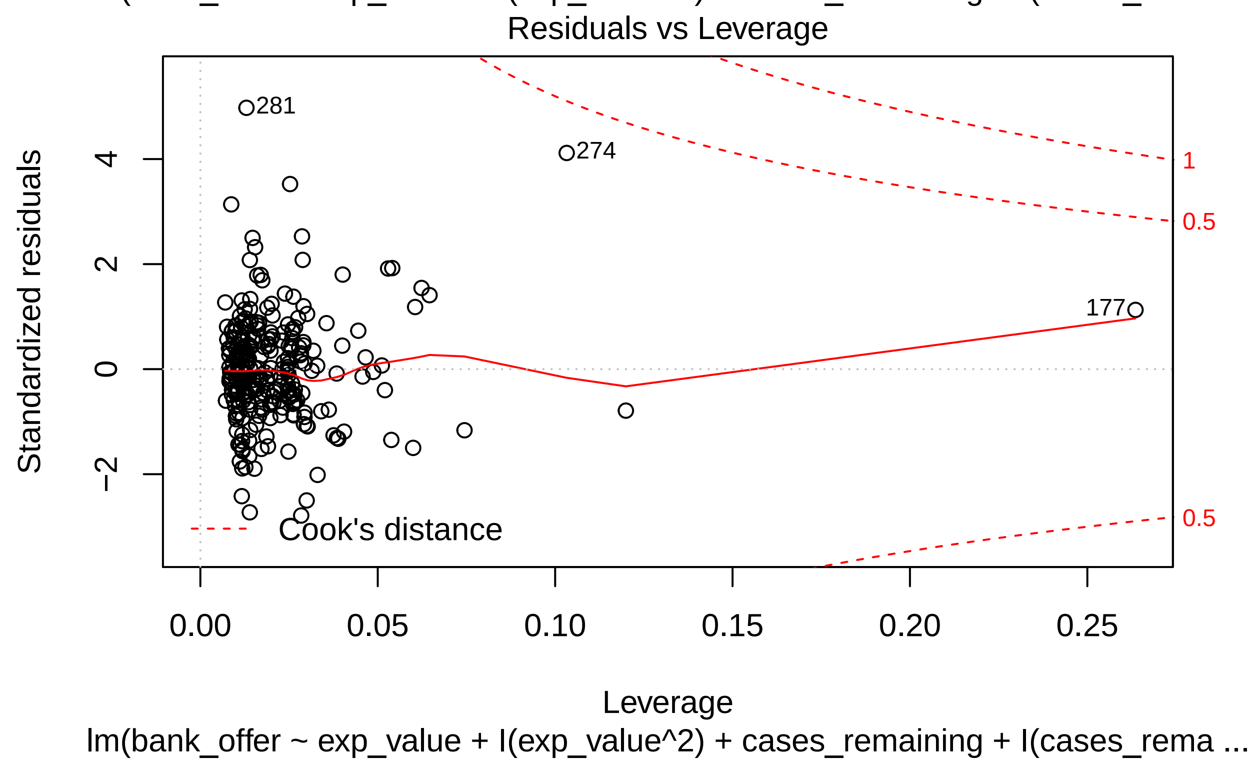
```
##
## Call:
## lm(formula = bank_offer ~ exp_value + I(exp_value^2) + cases_remaining +
##     I(cases_remaining^2) + max, data = train2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -70476 -15013  -1643  13020 126828
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.545e+04  7.057e+03   6.441 5.29e-10 ***
## exp_value             5.835e-01  7.653e-02   7.624 4.00e-13 ***
## I(exp_value^2)        8.298e-07  1.642e-07   5.052 7.97e-07 ***
## cases_remaining      -6.302e+03  1.529e+03  -4.123 4.96e-05 ***
## I(cases_remaining^2)  1.123e+02  6.009e+01   1.869   0.0627 .
## max                  -4.898e-02  1.044e-02  -4.691 4.29e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25660 on 275 degrees of freedom
## Multiple R-squared:  0.8967, Adjusted R-squared:  0.8948
## F-statistic: 477.2 on 5 and 275 DF,  p-value: < 2.2e-16
```

```
plot(online1)
```


Residuals vs Fitted
281
216
274
Residuals
150000
50000
0
−50000
0e+00
1e+05
2e+05
3e+05
4e+05
Fitted values
lm(bank_offer ~ exp_value + I(exp_value^2) + cases_remaining + I(cases_rema ...


Normal Q−Q
281
274
216
Standardized residuals
4
2
0
−2
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(bank_offer ~ exp_value + I(exp_value^2) + cases_remaining + I(cases_rema ...





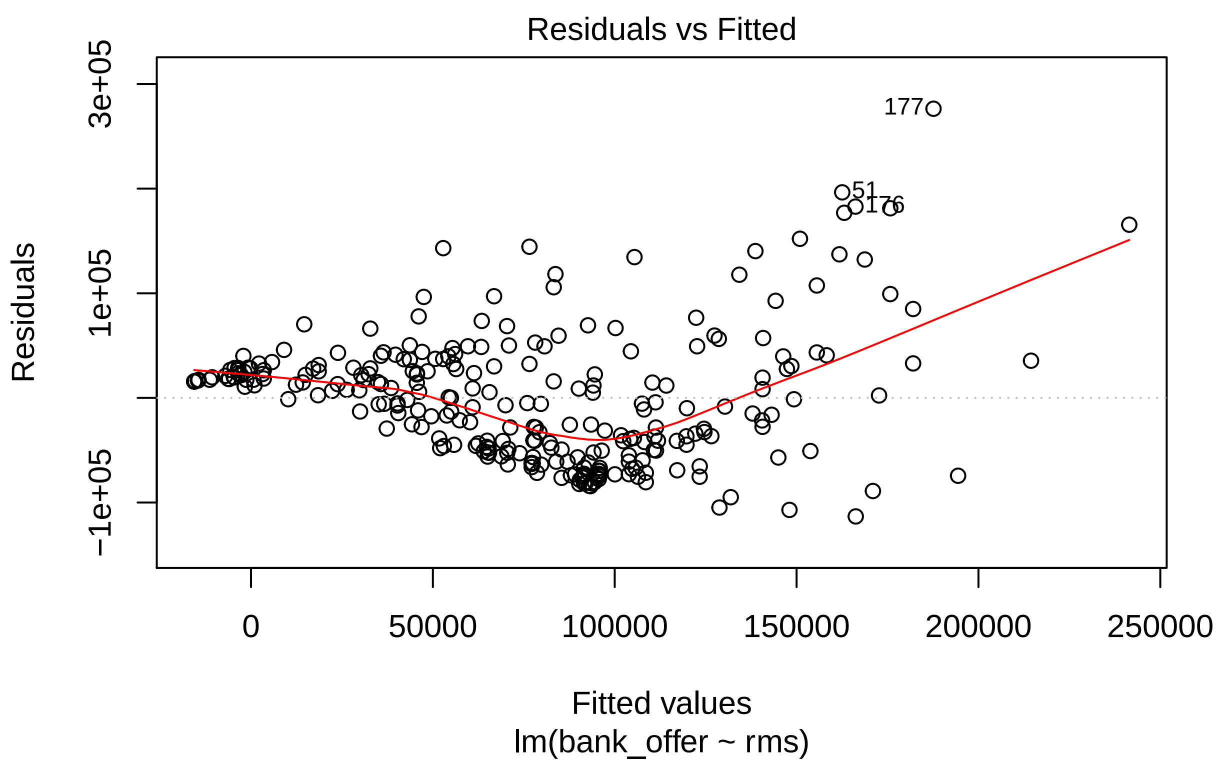
Online Model created for the british verison of the show

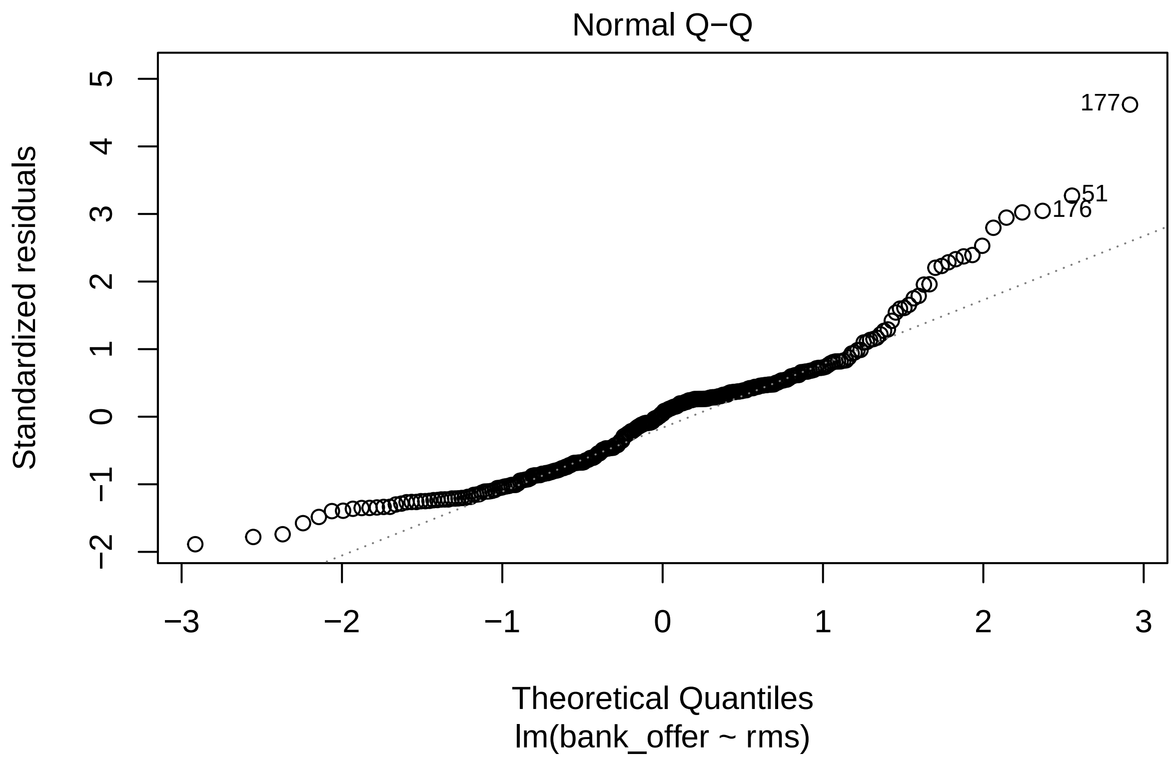
```
online2 <- lm(bank_offer ~ rms, train2)
summary(online2)
```

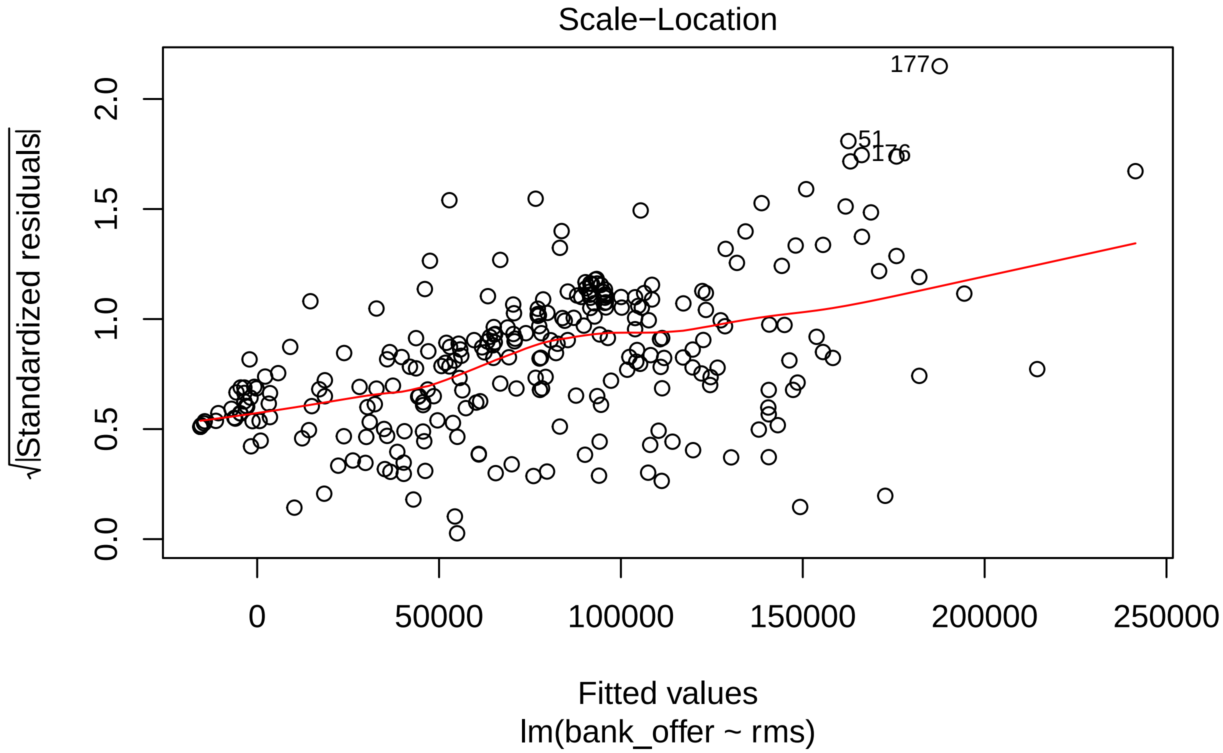
```
##
## Call:
```

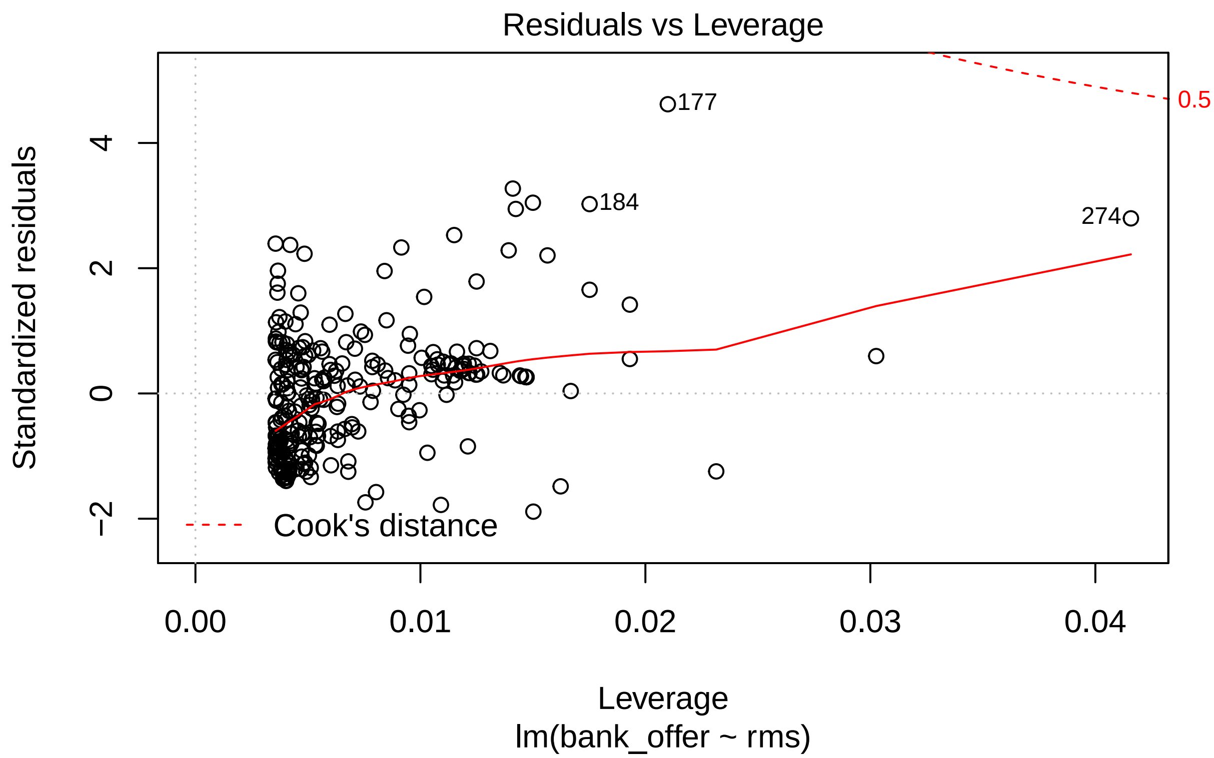
```
## lm(formula = bank_offer ~ rms, data = train2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -113271  -48329    2565   28532  276353
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.563e+04  7.337e+03   -2.13   0.0341 *
## rms          3.424e-01  2.421e-02   14.14   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60470 on 279 degrees of freedom
## Multiple R-squared:  0.4175, Adjusted R-squared:  0.4154
## F-statistic:   200 on 1 and 279 DF,  p-value: < 2.2e-16
```

```r
plot(online2)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(bank_offer ~ rms)
177
51
176


Scale−Location
√|Standardized residuals|
Fitted values
lm(bank_offer ~ rms)
177
51
176



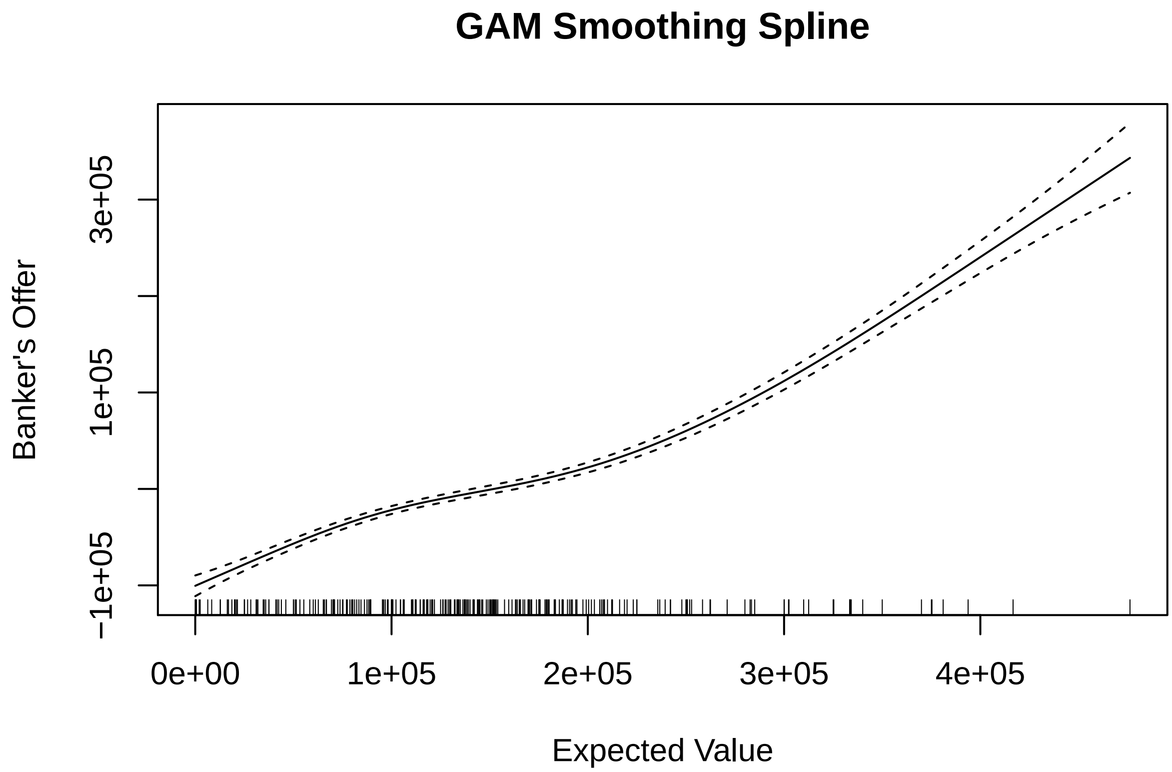
GAM model

```
modGAM <- gam(bank_offer ~ s(exp_value) + round, data=train2)
summary(modGAM)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## bank_offer ~ s(exp_value) + round
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    20376       4178   4.877 1.85e-06 ***
## round2         17133       5535   3.095  0.00217 **
## round3         37799       5566   6.791 7.18e-11 ***
## round4         55535       5699   9.744  < 2e-16 ***
## round5         66658       5872  11.352  < 2e-16 ***
## round6         84004       5998  14.005  < 2e-16 ***
## round7        100568       6351  15.836  < 2e-16 ***
## round8         90494       7177  12.608  < 2e-16 ***
## round9         84872       9679   8.769  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                edf Ref.df     F p-value
## s(exp_value) 4.643  5.756 345.5  <2e-16 ***
## ---
```

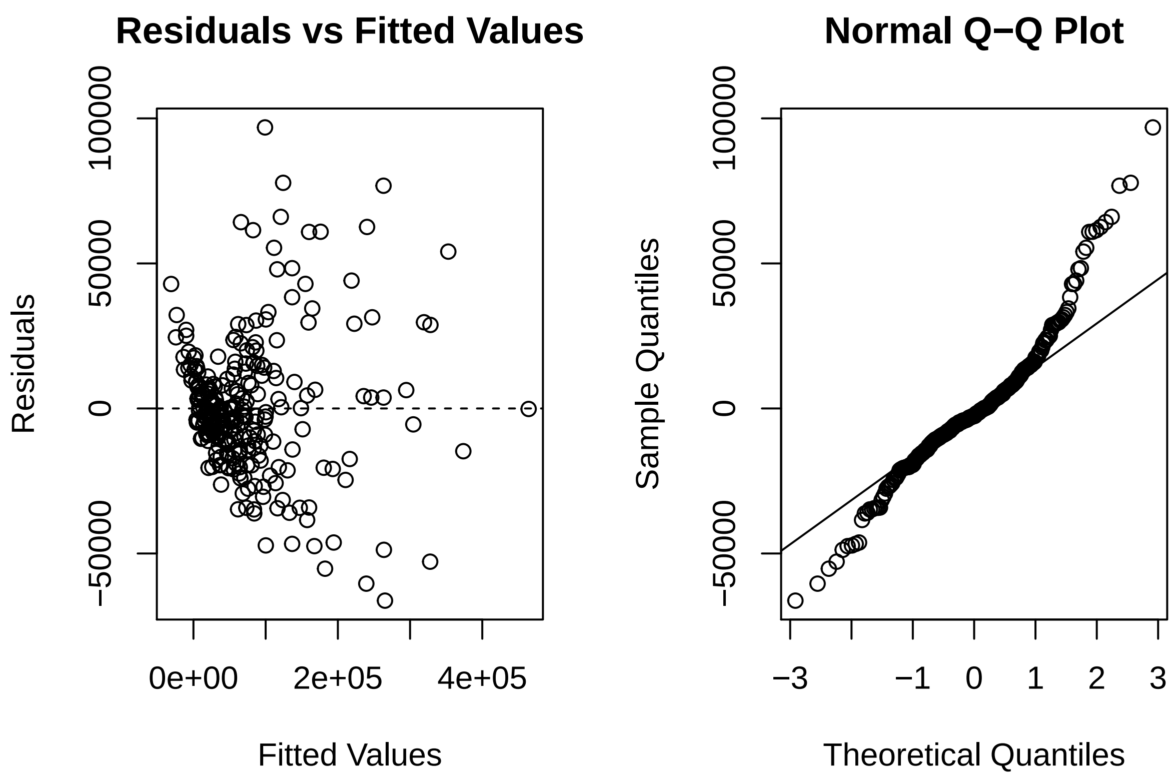
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.912   Deviance explained = 91.6%
## GCV = 5.8075e+08  Scale est. = 5.5256e+08  n = 281
```

```
plot(modGAM,main = "GAM Smoothing Spline", xlab ="Expected Value", ylab = "Banker's Offer")
```




```
res = residuals(modGAM, type="deviance") #compute the deviance residuals

#residual and QQ plot
par(mfrow=c(1,2))
plot(predict(modGAM, type = "link"), res, main = "Residuals vs Fitted Values", xlab ="Fitted Values", yl
abline(h=0, lty=2)
qqnorm(res)
qqline(res)
```



MSPE for linear model, GAM, online model 1 and online model 2

```
y = test2$bank_offer

y_hat1 = predict(lmod2, newdata = test2)
mspe1 = mean((y - y_hat1)^2); mspe1
```

```
## [1] 656184346
```

```
y_hat2 = predict(modGAM, newdata = test2)
mspe2 = mean((y - y_hat2)^2); mspe2
```

```
## [1] 404489740
```

```
y_hat3 = predict(online1, newdata = test2)
mspe3 = mean((y - y_hat3)^2); mspe3
```

```
## [1] 403511794
```

```
y_hat4 = predict(online2, newdata = test2)
mspe4 = mean((y - y_hat4)^2); mspe4
```

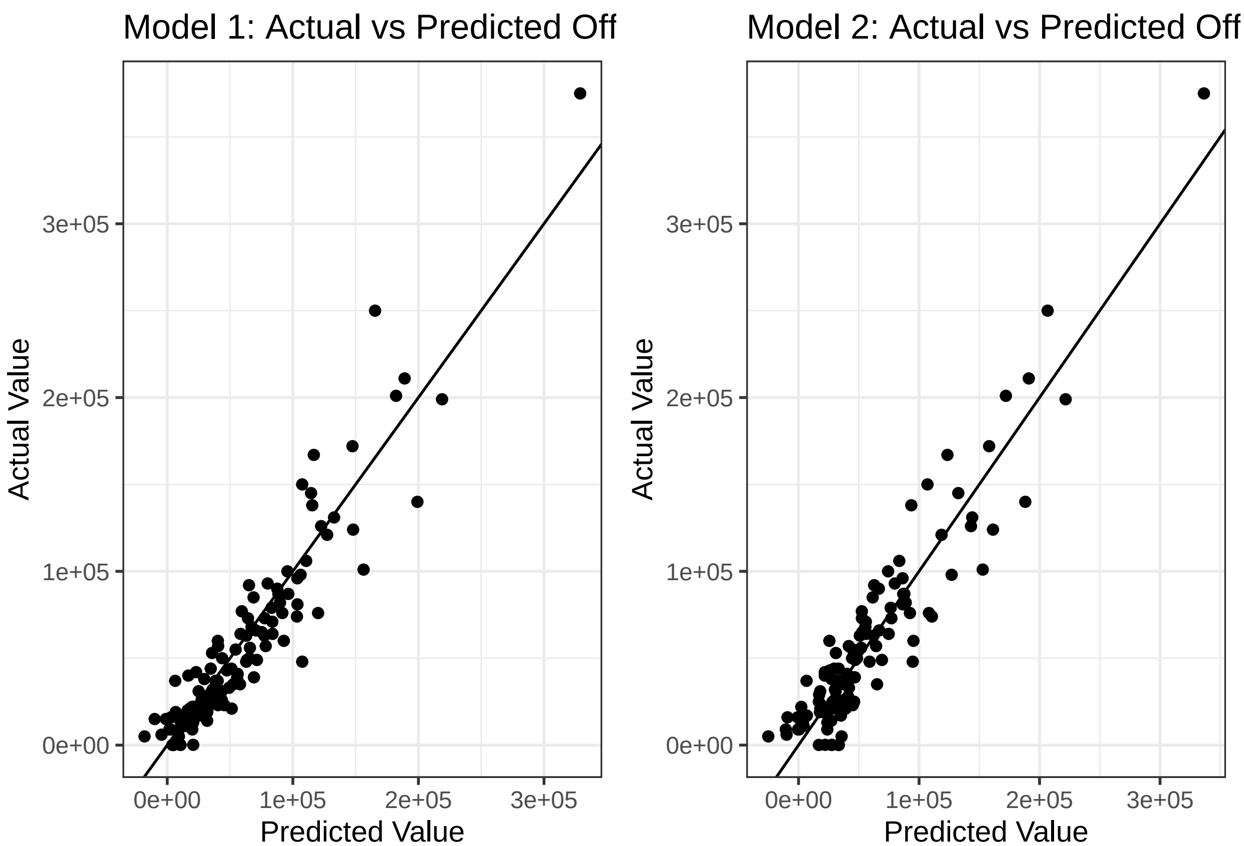
```
## [1] 3016252592
```

GAM predictions vs online model predictions

```
p1 <- ggplot(test2, aes(predict(modGAM, newdata = test2), bank_offer)) +
  geom_point() +
  geom_abline(slope=1) +
  xlab("Predicted Value") +
  ylab("Actual Value") +
  ggtitle("Model 1: Actual vs Predicted Offer") +
  theme_bw()
```

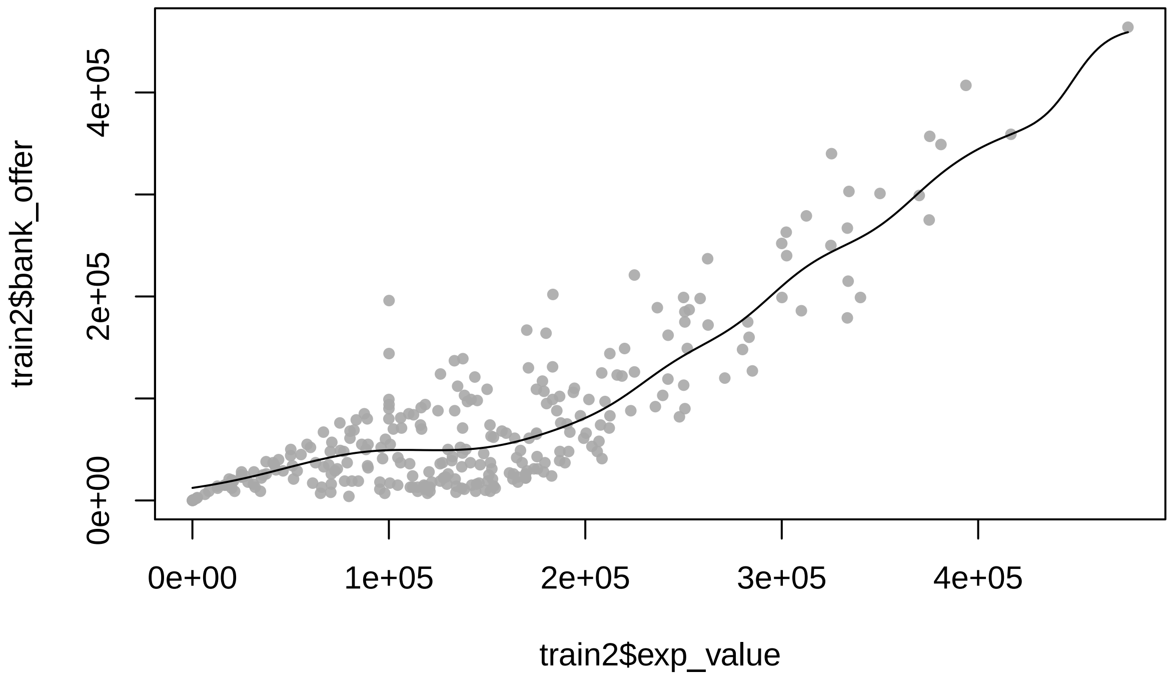
```r
p2 <- ggplot(test2, aes((predict(online1, newdata = test2)), bank_offer)) +
  geom_point() +
  geom_abline(slope=1) +
  xlab("Predicted Value") +
  ylab("Actual Value") +
  ggtitle("Model 2: Actual vs Predicted Offer") +
  theme_bw()

grid.arrange(p1, p2, ncol=2)
```

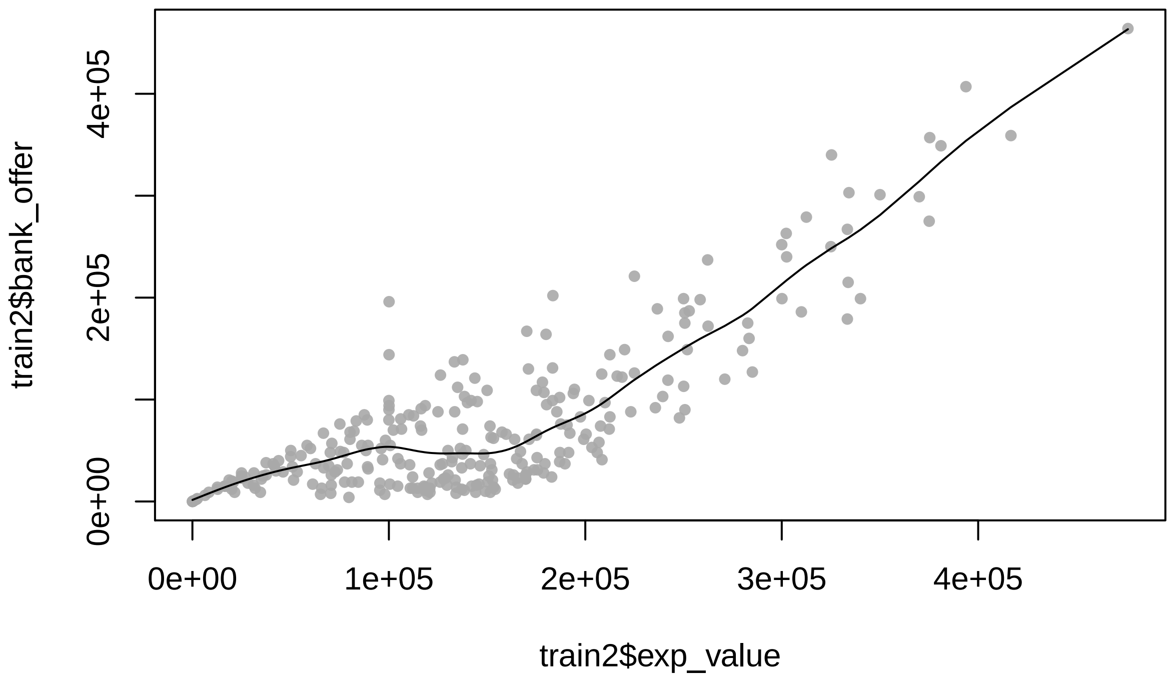


testing other non-parametric models

```r
plot(train2$bank_offer ~ train2$exp_value, pch = 16, cex = 0.8, col = alpha("darkgrey", 0.9))
lines(ksmooth(train2$exp_value, train2$bank_offer, kernel = "normal", bandwidth =  65000))
```

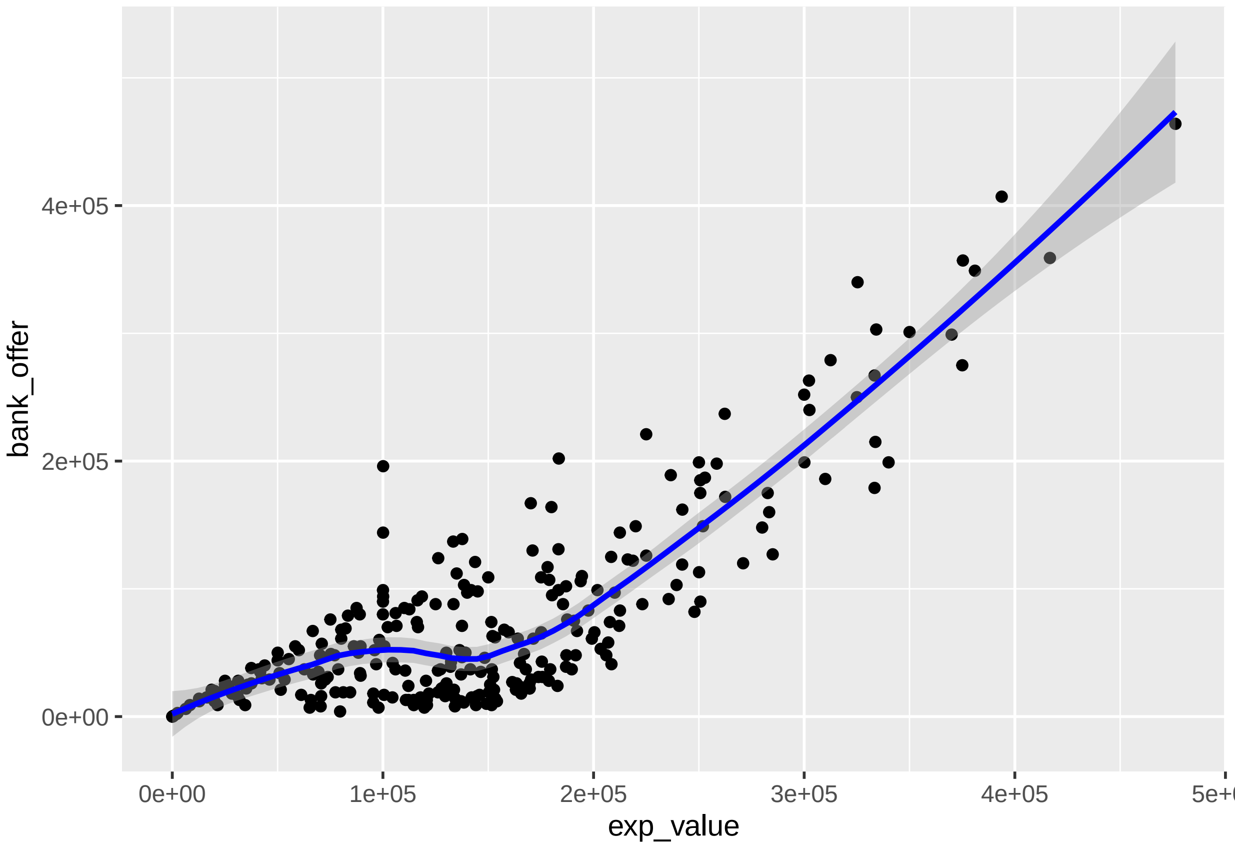



```r
plot(train2$bank_offer ~ train2$exp_value, pch = 16, cex = 0.8, col = alpha("darkgrey", 0.9))
lines(smooth.spline(train2$exp_value, train2$bank_offer, spar = 1.1))
```




```r
smooth_mod <- smooth.spline(train2$exp_value, train2$bank_offer, spar = 1.1)

ggplot(train2, aes(x = exp_value, y = bank_offer)) +
    geom_point() +
    geom_smooth(method = "loess", formula = "y ~ x", color = "blue", span = .5)
```




```
lr <- loess(train2$bank_offer ~ exp_value, train2, span = 0.5)
summary(lr)
```

```
## Call:
## loess(formula = train2$bank_offer ~ exp_value, data = train2,
##     span = 0.5)
##
## Number of Observations: 281
## Equivalent Number of Parameters: 7.36
## Residual Standard Error: 35840
## Trace of smoother matrix: 8.11  (exact)
##
## Control settings:
##   span     :  0.5
##   degree   :  2
##   family   :  gaussian
##   surface  :  interpolate      cell = 0.2
##   normalize:  TRUE
##  parametric:  FALSE
## drop.square:  FALSE
```

mspe for other non-parametric models, loess is the best smoothing spline but GAM overall best because I can include addtional variables

```
yhat_gam <- predict(modGAM, newdata = test2)
mspe1 = mean((y - yhat_gam)^2); mspe1
```

```
## [1] 404489740
```

```r
yhat_ks <- ksmooth(train2$bank_offer, train2$exp_value, kernel = "normal", 65000, x.points = test2$exp_v
mspe2 <- mean((y - yhat_ks$y)^2); mspe2
```

```
## [1] 24119268605
```

```r
yhat_ss <- predict(smooth_mod, x = test2$exp_value)
mspe3 <- mean((y - yhat_ss$y)^2); mspe3
```

```
## [1] 1193131918
```

```r
yhat_loess <- predict(lr, newdata = test2$exp_value)
mspe4 <- mean((y - yhat_loess)^2); mspe4
```

```
## [1] 1186574291
```

```r
min(mspe1, mspe2, mspe3, mspe4)
```

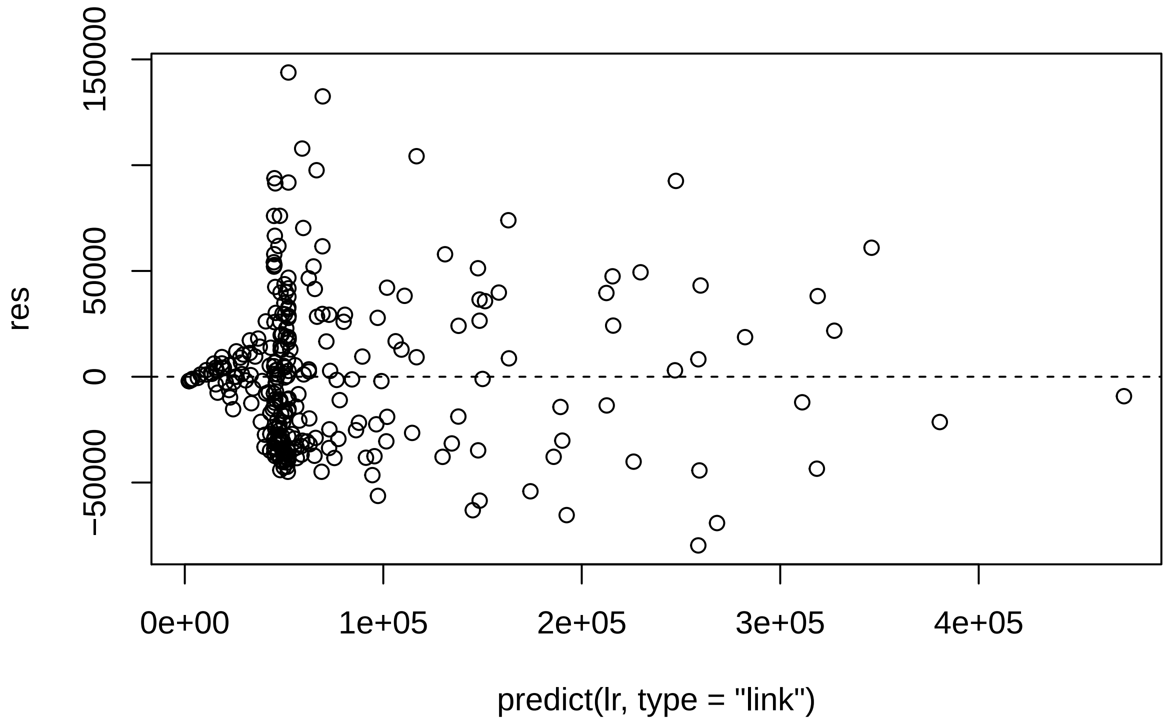
```
## [1] 404489740
```

loess residuals

```r
res <- residuals(lr, type="deviance")
summary(lr)
```

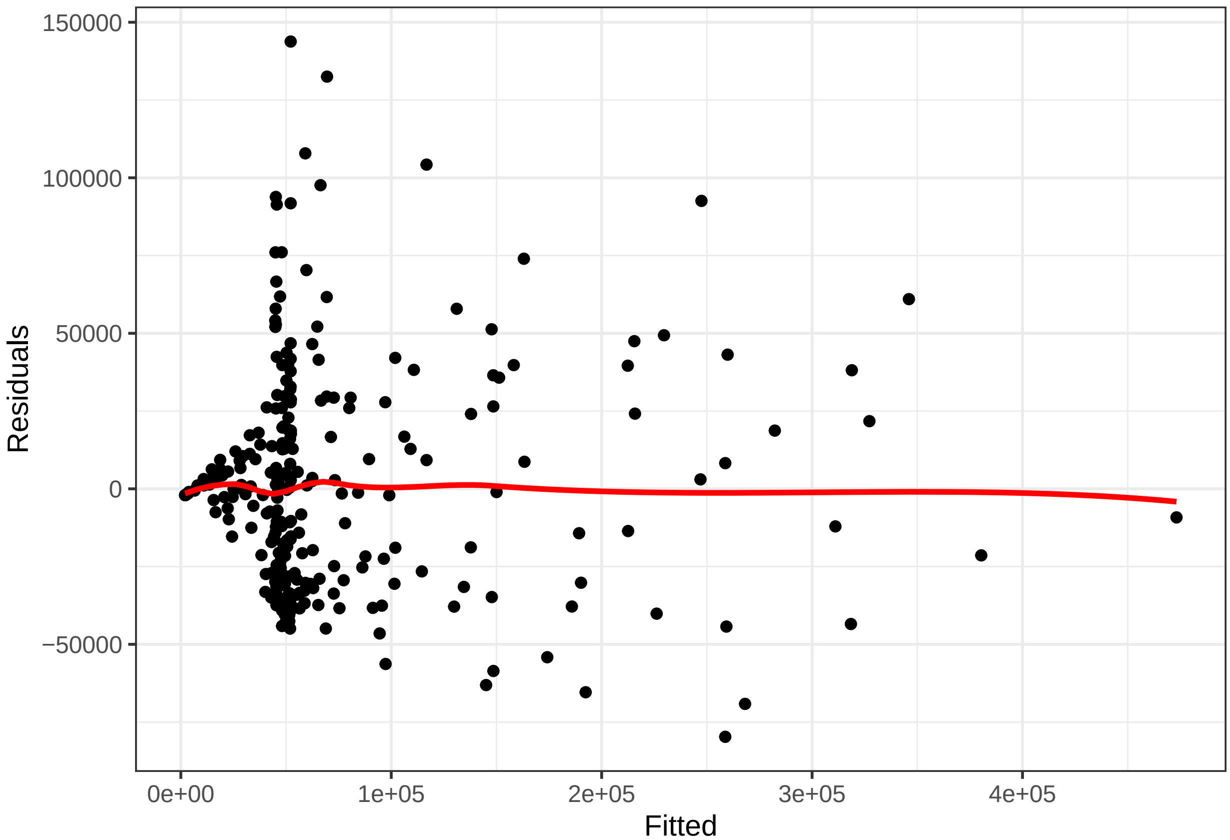
```
## Call:
## loess(formula = train2$bank_offer ~ exp_value, data = train2,
##     span = 0.5)
##
## Number of Observations: 281
## Equivalent Number of Parameters: 7.36
## Residual Standard Error: 35840
## Trace of smoother matrix: 8.11  (exact)
##
## Control settings:
##   span     :  0.5
##   degree   :  2
##   family   :  gaussian
##   surface  :  interpolate      cell = 0.2
##   normalize:  TRUE
##  parametric:  FALSE
## drop.square:  FALSE
```

```r
plot(predict(lr, type = "link"), res)
abline(h=0, lty=2)
```




```
df <- data.frame(predict(lr, type = "link"), res)
colnames(df)[1] <- "x"

ggplot(df, aes(x = x, y = res)) +
    geom_point() +
    xlab("Fitted") +
    geom_smooth(method = "loess", formula = 'y ~ x', se = F, col = "red") +
    ylab("Residuals") +
    theme_bw()
```



150000
100000
50000
0
−50000
Residuals
0e+00
1e+05
2e+05
3e+05
4e+05
Fitted