

# Predicting Pitcher DL

*Roger Chow*

*March 19, 2017*

```
setwd("F:/Capstone_Workspace/predictDL/");
library('RODBC');

## Warning: package 'RODBC' was built under R version 3.2.5
library('DBI');
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('stringi');
library('sqldf');

## Warning: package 'sqldf' was built under R version 3.2.5
## Loading required package: gsubfn
## Warning: package 'gsubfn' was built under R version 3.2.5
## Loading required package: proto
## Loading required package: RSQLite
## Warning: package 'RSQLite' was built under R version 3.2.5
library('corrplot');

## Warning: package 'corrplot' was built under R version 3.2.5
library('reshape2');
library('ggplot2');

## Warning: package 'ggplot2' was built under R version 3.2.5
library('caret');

## Warning: package 'caret' was built under R version 3.2.5
## Loading required package: lattice
dbhandle <- odbcDriverConnect('driver={SQL Server};server=localhost;database=PitchFx;trusted_connection=true')
query <-
'SELECT m.rsid, min(ms.nameLast) as nameLast, min(ms.nameFirst) as nameFirst,
year(p.GameDate) as season,
avg(p.x) as x,
avg(p.y) as y,
```

```

        avg(p.start_speed) as start_speed,
        avg(p.end_speed) as end_speed,
        avg(p.sz_top) as sz_top,
        avg(p.sz_bot) as sz_bot,
        avg(p.pfx_x) as pfx_x,
        avg(p.pfx_z) as pfx_z,
        avg(p.px) as px,
        avg(p.pz) as pz,
        avg(p.x0) as x0,
        avg(p.y0) as y0,
        avg(p.z0) as z0,
        avg(p.vx0) as vx0,
        avg(p.vy0) as vy0,
        avg(p.vz0) as vz0,
        avg(p.ax) as ax,
        avg(p.ay) as ay,
        avg(p.az) as az,
        avg(p.break_y) as break_y,
        avg(p.break_angle) as break_angle,
        avg(p.break_length) as break_length,
        avg(p.spin_dir) as spin_dir,
        avg(p.spin_rate) as spin_rate,
        sum(p.num_pitches) as num_pitches
FROM [PitchFx].[dbo].[GamesAtBatsAggregatePitches] p
INNER JOIN [Mapping].[dbo].[RSID_MLBID_MAP] m on p.pitcher = m.mlbid
INNER JOIN [Lahman].[dbo].[Master] ms on ms.retroid = m.rsid
GROUP BY m.rsid, year(p.GameDate)'

pitches <- sqlQuery(dbhandle, query);
pitches <- pitches[complete.cases(pitches),];
close(dbhandle);

dbhandle <- odbcDriverConnect('driver={SQL Server};server=localhost;database=PitchFx;trusted_connection=1');
query <- "
SELECT rsid, 2011 as season_dl, sum(days) as DLDays
FROM [DisabledList].[dbo].[DL2011]
WHERE Position in ('LHP','RHP','RP','SP','P')
GROUP BY rsid
UNION
SELECT rsid, 2012 as season_dl, sum(days) as DLDays
FROM [DisabledList].[dbo].[DL2012]
WHERE Pos in ('LHP','RHP','RP','SP','P')
GROUP BY rsid
UNION
SELECT rsid, 2013 as season_dl, sum(days) as DLDays
FROM [DisabledList].[dbo].[DL2013]
WHERE Position in ('LHP','RHP','RP','SP','P')
GROUP BY rsid
UNION
SELECT rsid, 2014 as season_dl, sum(days) as DLDays
FROM [DisabledList].[dbo].[DL2014]
WHERE Position in ('LHP','RHP','RP','SP','P')
GROUP BY rsid
UNION

```

```

SELECT rsid, 2015 as season_dl, sum(days) as DLDays
    FROM [DisabledList].[dbo].[DL2015]
    WHERE Position in ('LHP','RHP','RP','SP','P')
    GROUP BY rsid
UNION
SELECT rsid, 2016 as season_dl, sum(days) as DLDays
    FROM [DisabledList].[dbo].[DL2016]
    WHERE Position in ('LHP','RHP','RP','SP','P')
    GROUP BY rsid
";

dl <- sqlQuery(dbhandle, query);
dl <- dl[complete.cases(dl),];
dl$season_1 <- dl$season-1;
close(dbhandle);

#use previous season to predict DL in current season
pitches_dl <- merge(x=pitches, y=dl, by.x=c("rsid", "season"), by.y=c("rsid", "season_1"), all.x = TRUE)

pitches_dl[pitches_dl==""] <- NA; #replace blanks with NA

pitches_dl$DLDays[is.na(pitches_dl$DLDays)] <- 0; #no DL pitchers are on DL for 0 days

drops <- c("season_dl");
pitches_dl <- pitches_dl[ , !(names(pitches_dl) %in% drops)];

pitches_dl <- pitches_dl[complete.cases(pitches_dl),];

pitches_dl_dataset <- pitches_dl[pitches_dl$season < 2016,]; #for modeling
pitches_dl_predict <- pitches_dl[pitches_dl$season == 2016,]; #for 2017 prediction

pitches_dl_dataset$OnDL <- as.factor(ifelse(pitches_dl_dataset$DLDays>0, 'YES', 'NO'));

summary(pitches_dl_dataset);

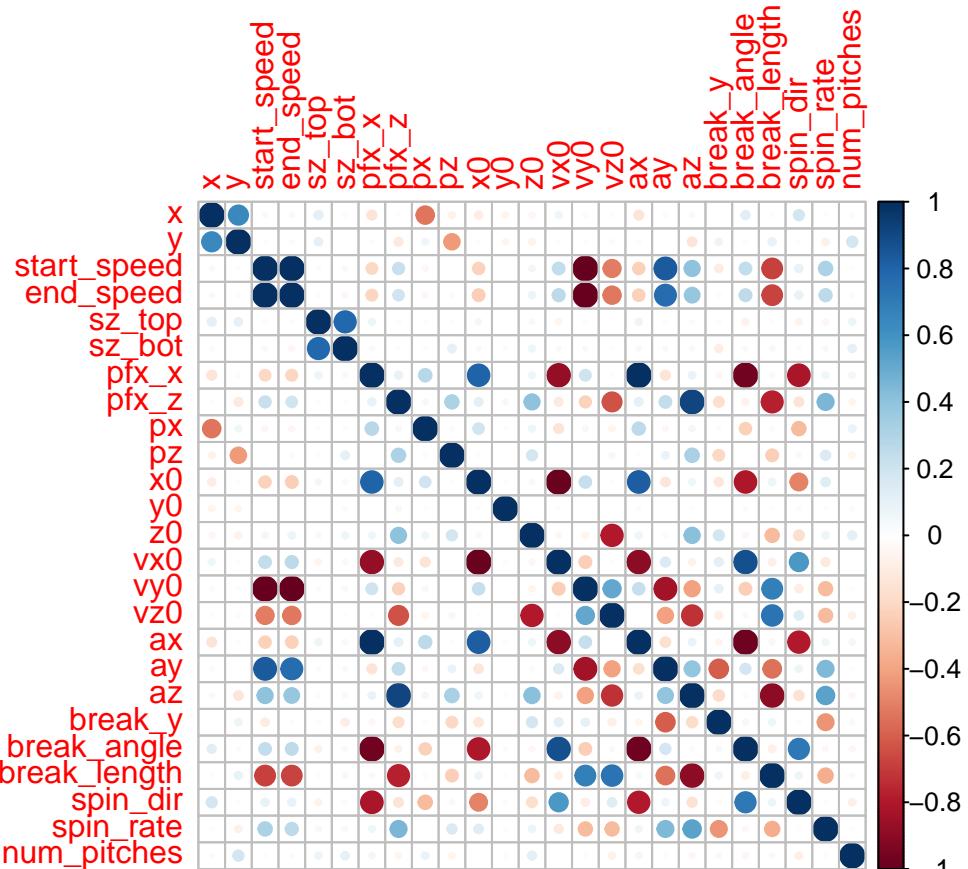
##          rsid        season      nameLast      nameFirst
## abadf001: 6   Min.   :2010  Rodriguez: 37   Chris   : 97
## adamm001: 6   1st Qu.:2011   Perez    : 27   Matt    : 92
## affej001: 6   Median  :2012  Hernandez: 26   Mike    : 72
## albem001: 6   Mean    :2012  Ramirez  : 25   Scott   : 69
## andeb004: 6   3rd Qu.:2014   Smith    : 24   David   : 68
## arrij001: 6   Max.    :2015  Johnson  : 23   Josh    : 67
## (Other) :4291                  (Other)  :4165  (Other):3862
##          x              y      start_speed     end_speed
##  Min.   : 64.74   Min.   :106.9   Min.   :53.77   Min.   :49.73
##  1st Qu.: 98.95   1st Qu.:143.7   1st Qu.:85.79   1st Qu.:79.19
##  Median :102.68   Median :147.2   Median :87.79   Median :81.05
##  Mean   :103.84   Mean   :148.9   Mean   :87.57   Mean   :80.80
##  3rd Qu.:107.32   3rd Qu.:150.9   3rd Qu.:89.74   3rd Qu.:82.77
##  Max.   :148.81   Max.   :191.6   Max.   :96.87   Max.   :89.96
##
##          sz_top       sz_bot      pfx_x      pfx_z
##  Min.   :0.000   Min.   :0.000   Min.   :-11.598  Min.   :-7.326
##  1st Qu.:3.391   1st Qu.:1.561   1st Qu.:-4.294  1st Qu.: 3.620

```

```

## Median :3.416   Median :1.584   Median : -2.173   Median : 5.059
## Mean   :3.412   Mean   :1.585   Mean   : -1.165   Mean   : 4.923
## 3rd Qu.:3.441   3rd Qu.:1.606   3rd Qu.:  2.050   3rd Qu.: 6.325
## Max.    :3.737   Max.    :1.820   Max.    : 11.167   Max.    :13.316
##
##          px           pz           x0           y0
## Min.   :-1.43400   Min.   :1.097   Min.   :-4.3108   Min.   :50
## 1st Qu.:-0.18681   1st Qu.:2.224   1st Qu.:-2.0329   1st Qu.:50
## Median :-0.08090   Median :2.339   Median :-1.3939   Median :50
## Mean   :-0.08393   Mean   :2.347   Mean   :-0.7184   Mean   :50
## 3rd Qu.: 0.02757   3rd Qu.:2.460   3rd Qu.: 1.0714   3rd Qu.:50
## Max.    : 1.02925   Max.    :3.727   Max.    : 5.2931   Max.    :50
##
##          z0           vx0          vy0          vz0
## Min.   :1.959   Min.   :-14.576   Min.   :-141.76   Min.   :-9.631
## 1st Qu.:5.655   1st Qu.:-4.067   1st Qu.:-131.32   1st Qu.:-5.305
## Median :5.906   Median : 4.480   Median :-128.48   Median :-4.450
## Mean   :5.866   Mean   : 2.066   Mean   :-128.16   Mean   :-4.258
## 3rd Qu.:6.148   3rd Qu.: 6.155   3rd Qu.:-125.58   3rd Qu.:-3.486
## Max.    :7.287   Max.    :11.062   Max.    :-78.81    Max.    : 9.976
##
##          ax           ay           az           break_y
## Min.   :-20.400  Min.   :10.34    Min.   :-43.122  Min.   :23.66
## 1st Qu.:-7.991   1st Qu.:25.39    1st Qu.:-24.924  1st Qu.:23.78
## Median :-4.285   Median :26.82    Median :-22.620   Median :23.80
## Mean   :-2.338   Mean   :26.83    Mean   :-22.778   Mean   :23.80
## 3rd Qu.: 3.734   3rd Qu.:28.36    3rd Qu.:-20.358  3rd Qu.:23.82
## Max.    :20.249  Max.    :35.35    Max.    :-7.535   Max.    :23.92
##
##          break_angle      break_length      spin_dir      spin_rate
## Min.   :-50.123   Min.   : 3.004   Min.   :102.6   Min.   : 688.8
## 1st Qu.:-9.559   1st Qu.: 5.820   1st Qu.:165.5   1st Qu.:1577.0
## Median :10.904   Median : 6.500   Median :182.0   Median :1743.1
## Mean   : 5.459   Mean   : 6.586   Mean   :183.0   Mean   :1737.9
## 3rd Qu.:18.440   3rd Qu.: 7.222   3rd Qu.:200.0   3rd Qu.:1906.7
## Max.    :51.992   Max.   :16.967   Max.   :297.6   Max.   :3024.0
##
##          num_pitches      DLDays      OnDL
## Min.    : 1.0   Min.   : 0.0   NO :3328
## 1st Qu.:217.5  1st Qu.: 0.0   YES: 999
## Median : 738.0  Median : 0.0
## Mean   :1045.1  Mean   :15.5
## 3rd Qu.:1325.0 3rd Qu.: 0.0
## Max.   :4441.0  Max.   :200.0
##
numeric_dataset <- pitches_dl_dataset[sapply(pitches_dl_dataset, is.numeric)];
#exclude season and DLDays
numeric_dataset <- numeric_dataset[2:(ncol(numeric_dataset)-1)];
m <- cor(numeric_dataset);
corrplot(m);

```



```

highlyCorrelated <- findCorrelation(m, cutoff=0.5);
lowCorrelatedCols <- colnames(numeric_dataset[-highlyCorrelated]);
print(lowCorrelatedCols);

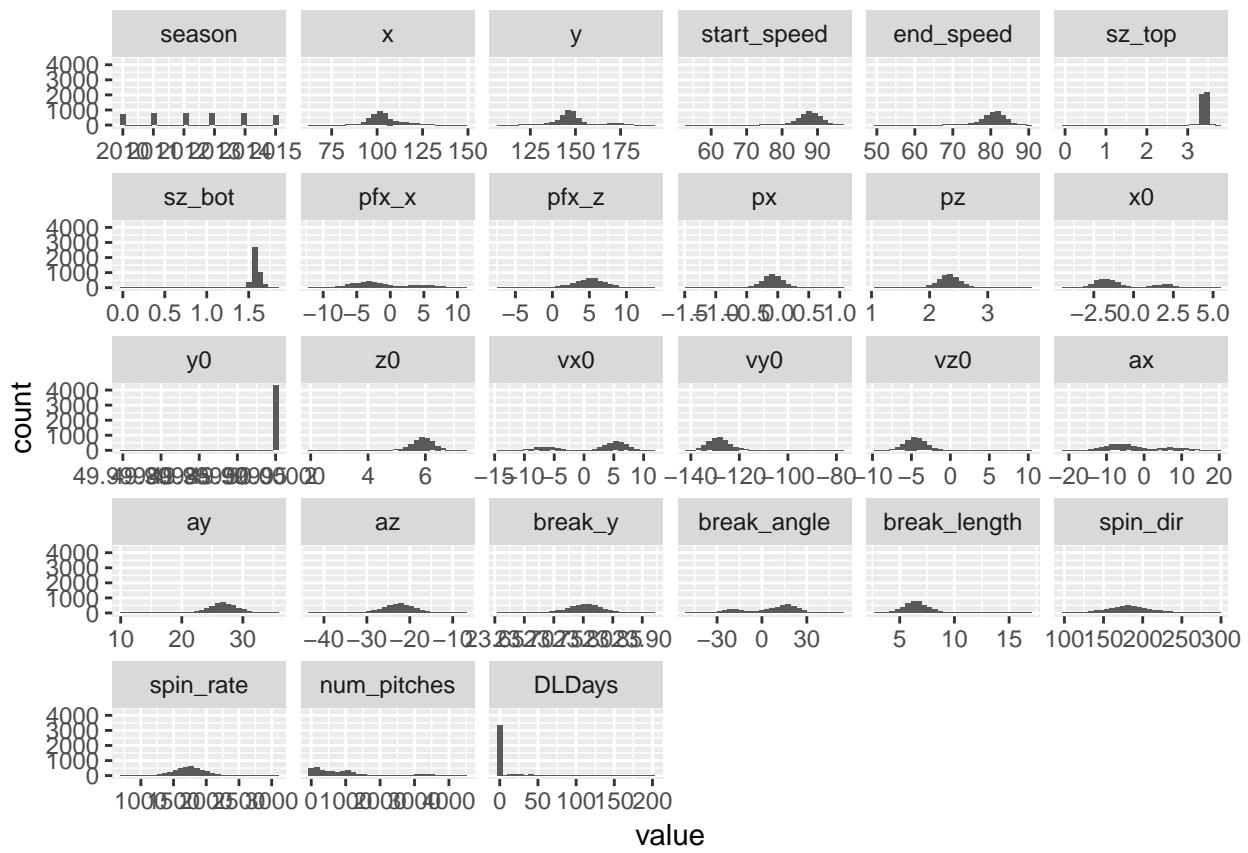
## [1] "y"                 "sz_bot"            "pfx_z"             "pz"                "x0"
## [6] "y0"               "z0"                "break_y"           "spin_dir"          "spin_rate"
## [11] "num_pitches"

d <- melt(pitches_dl_dataset[sapply(pitches_dl_dataset, is.numeric)]);

## No id variables; using all as measure variables
ggplot(d,aes(x = value)) + facet_wrap(~variable,scales = "free_x") + geom_histogram();

## `stat_bin()` using `bins = 30` . Pick better value with `binwidth` .

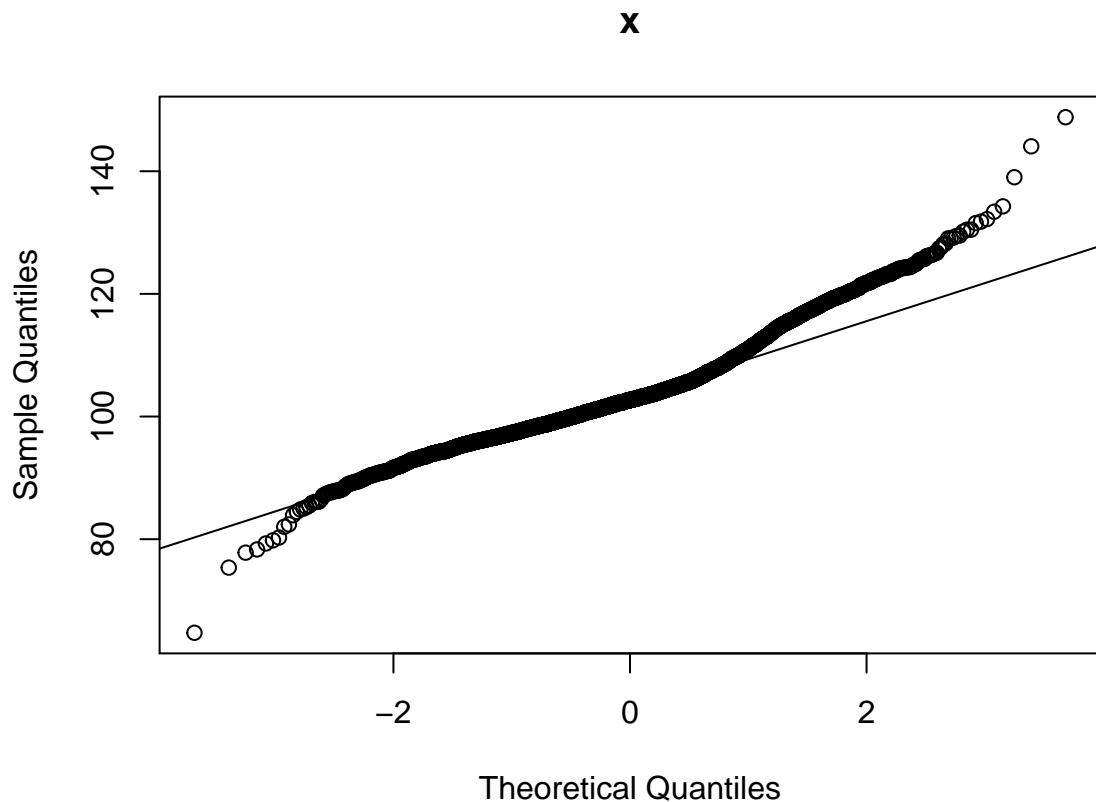
```

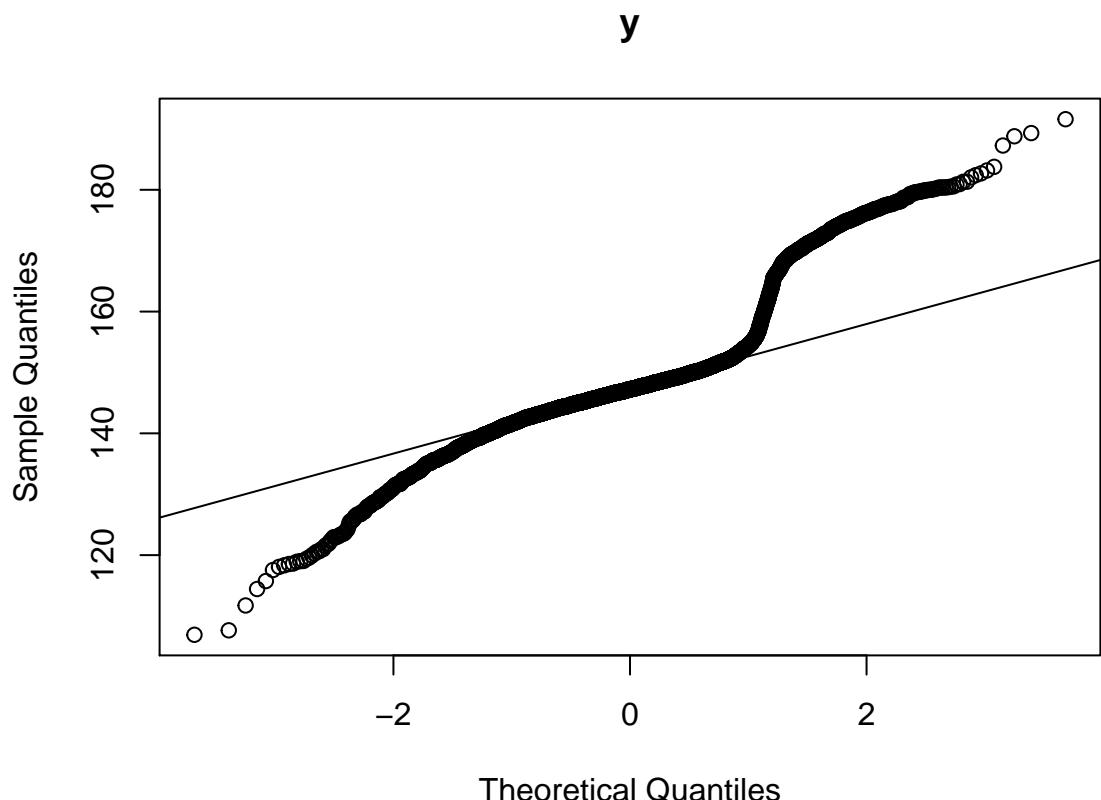


```

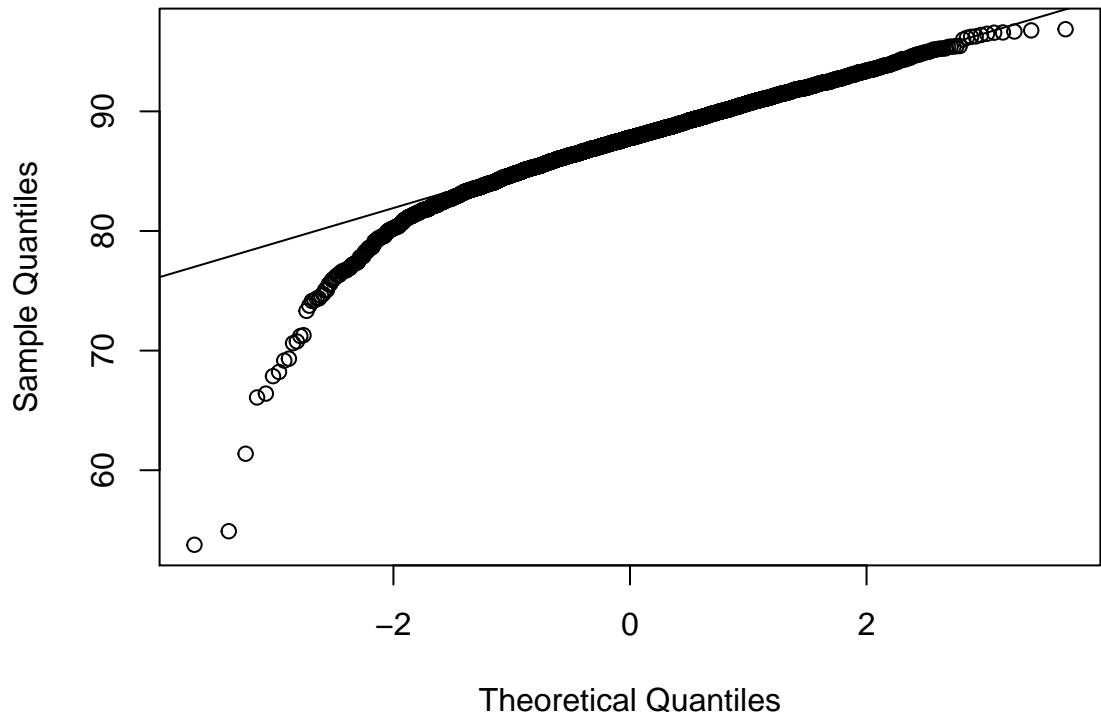
par(mar=c(4,4,4,4))
#for (i in 5:ncol(pitches_dl_dataset[,1: ncol(pitches_dl_dataset) - 1 ])){
for (i in 5:(ncol(pitches_dl_dataset)-2)){
  tmp <- pitches_dl_dataset[, i];
  qqnorm(tmp, main = colnames(pitches_dl_dataset[i]));
  qqline(tmp);
}

```

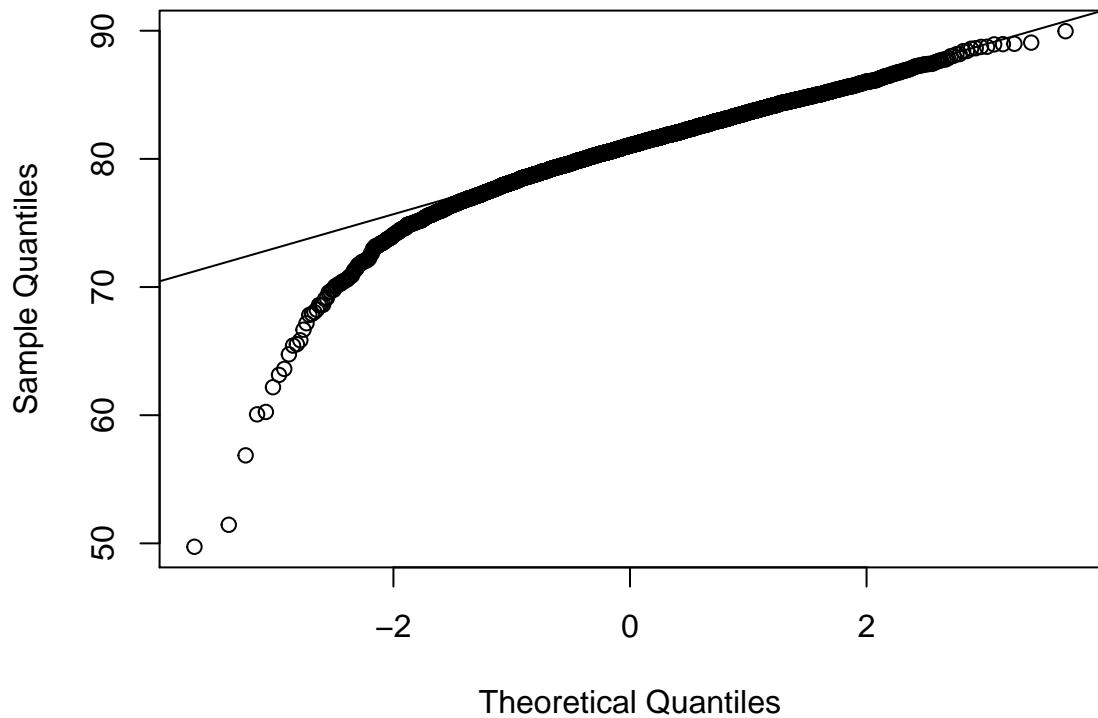




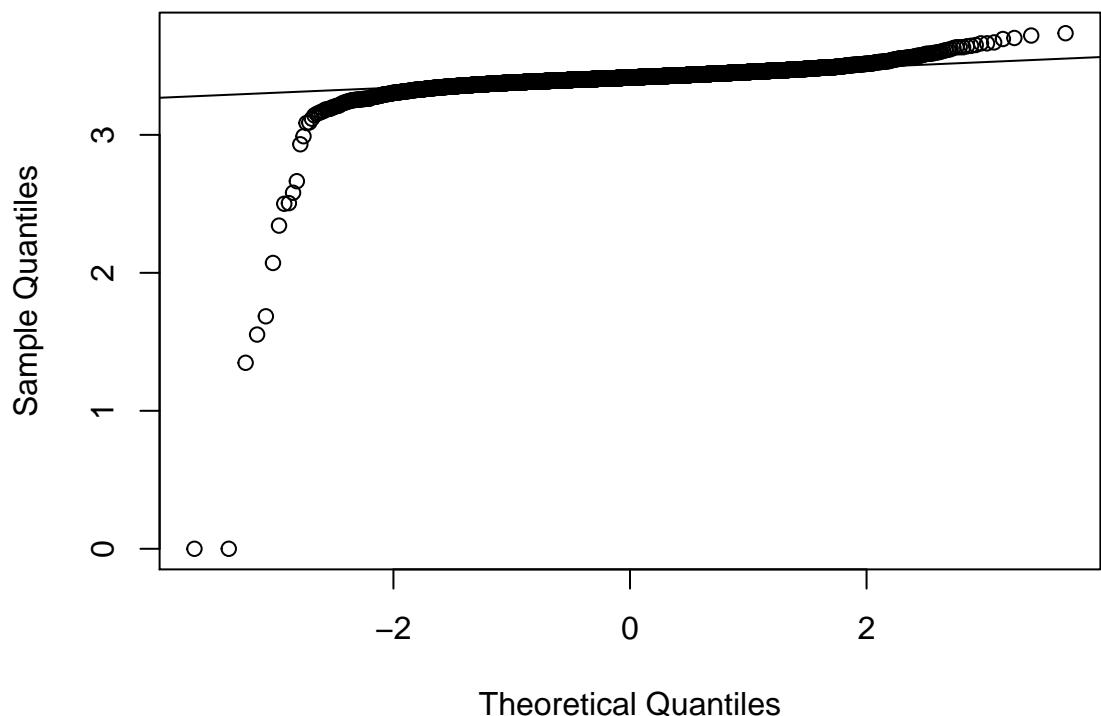
### **start\_speed**



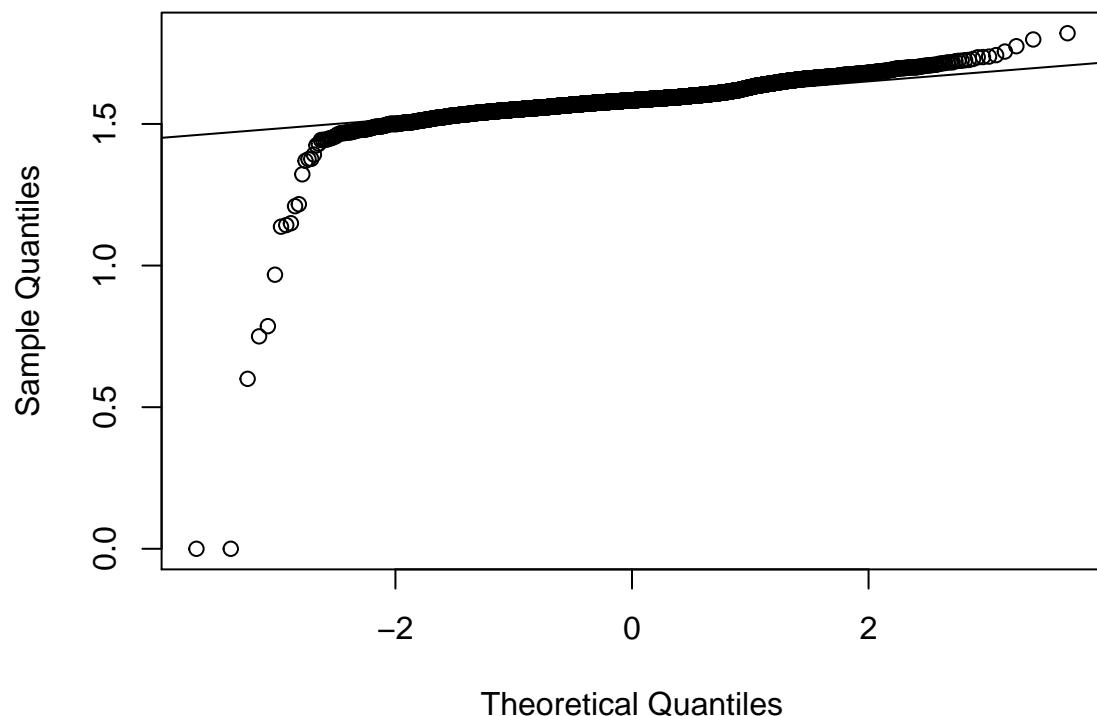
**end\_speed**



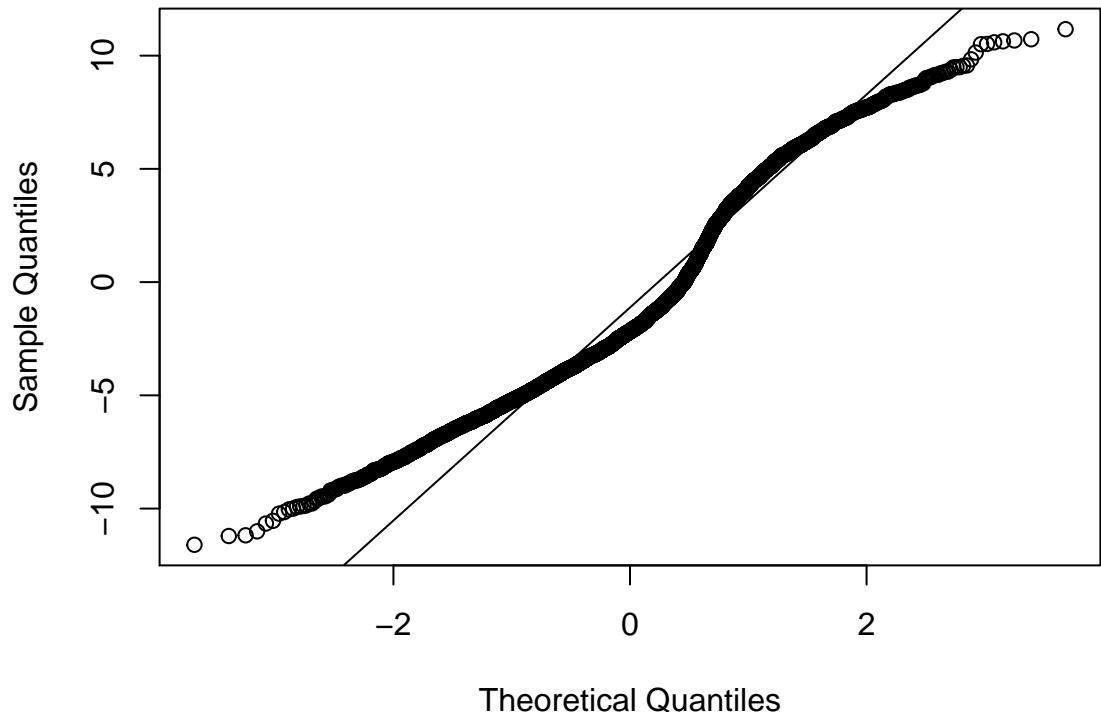
**sz\_top**



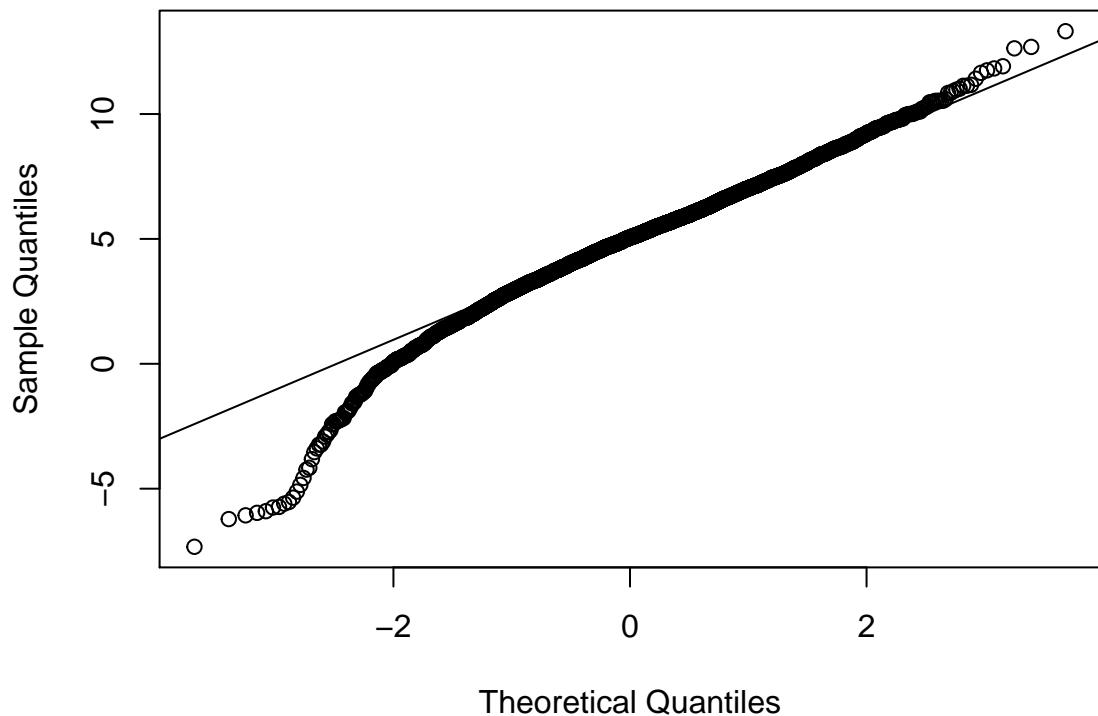
**sz\_bot**

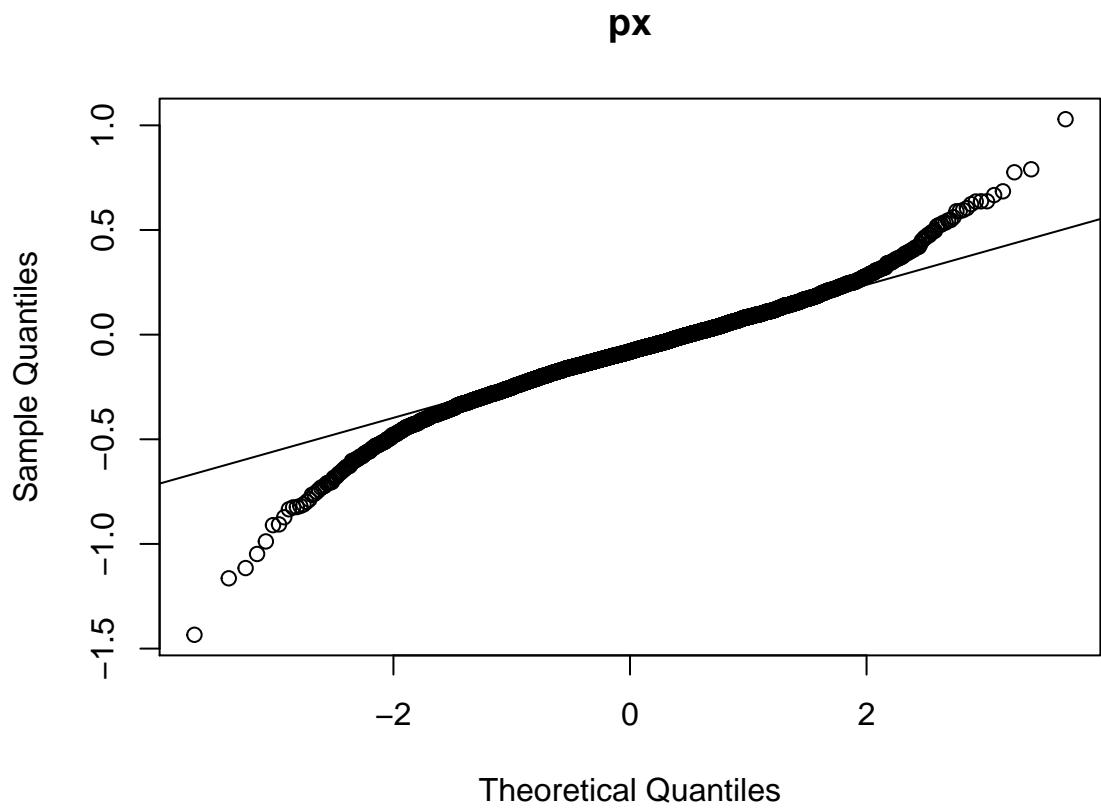


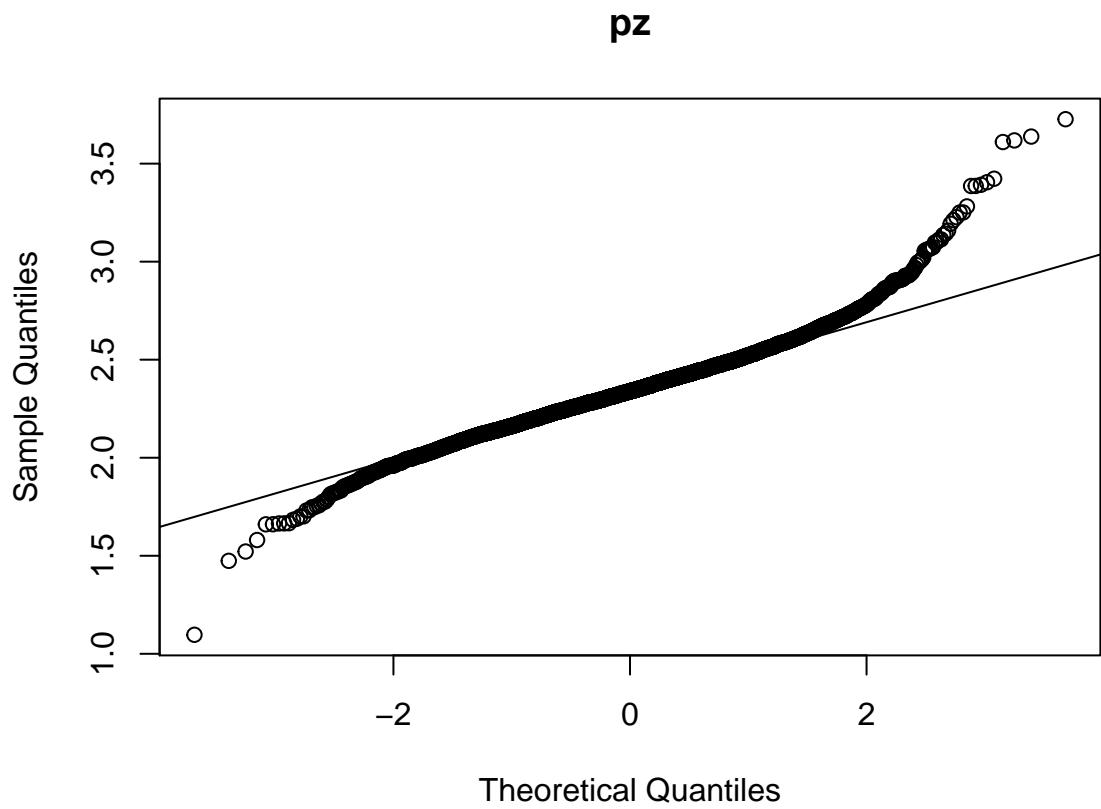
**pfx\_x**

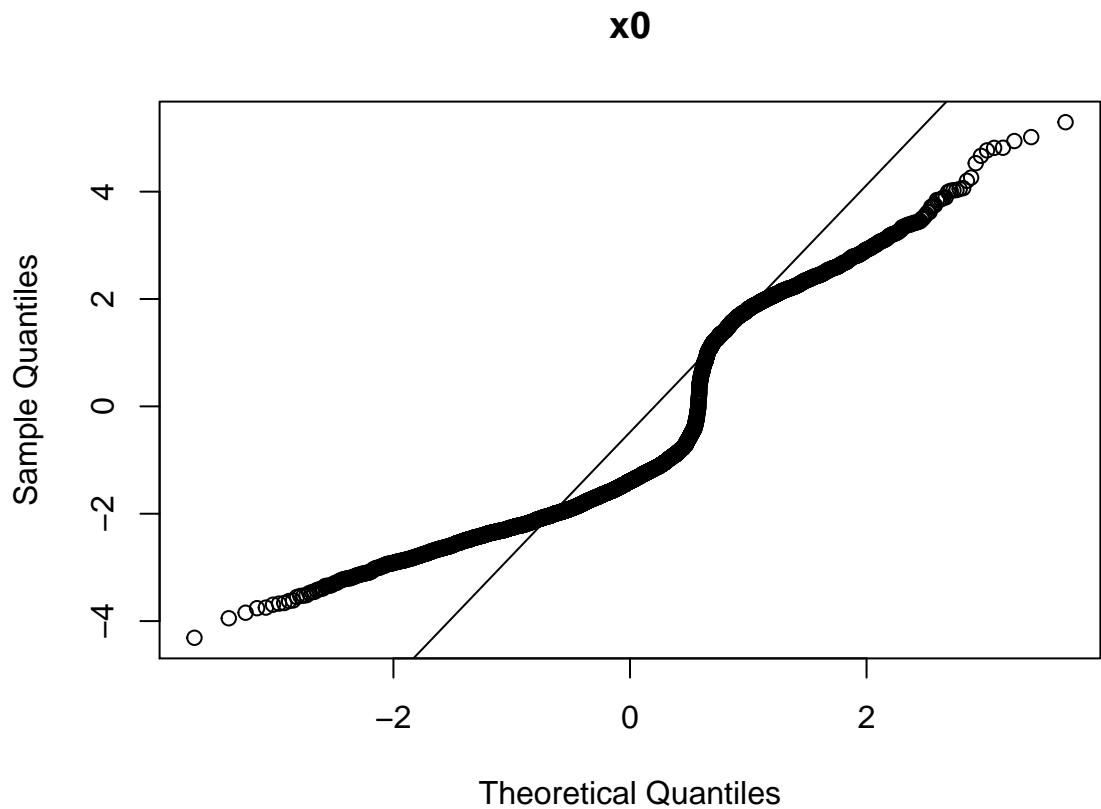


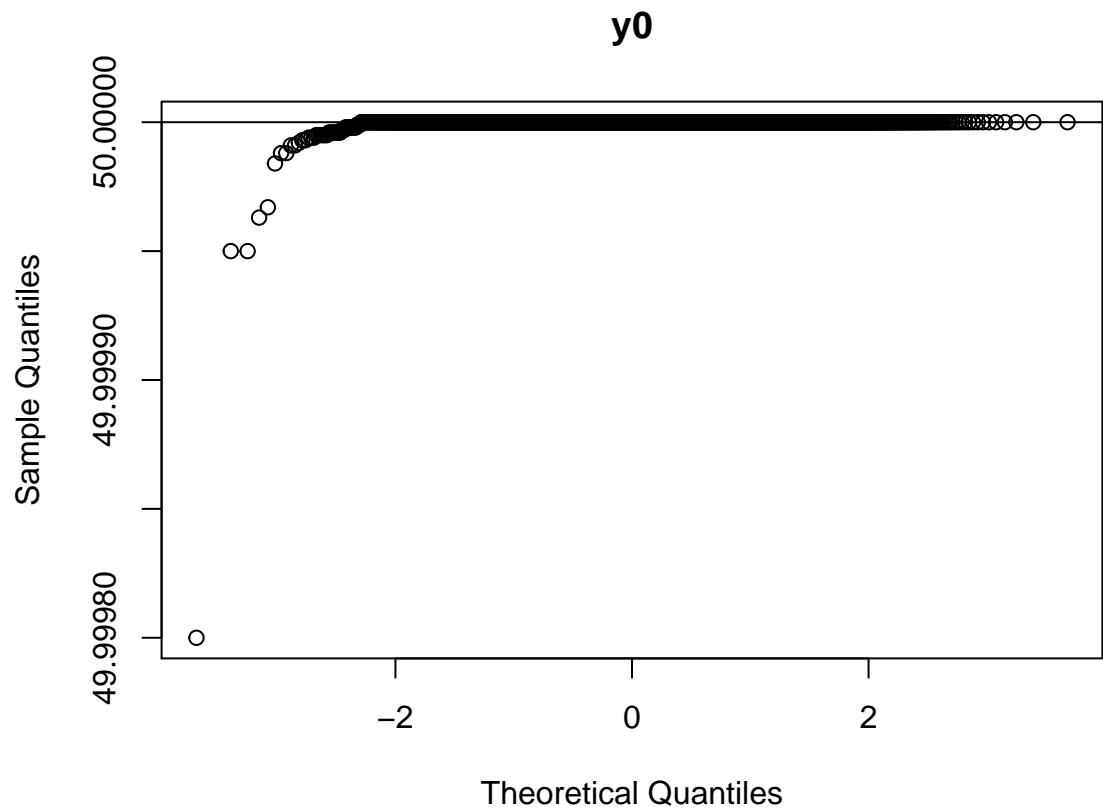
**pxf\_z**

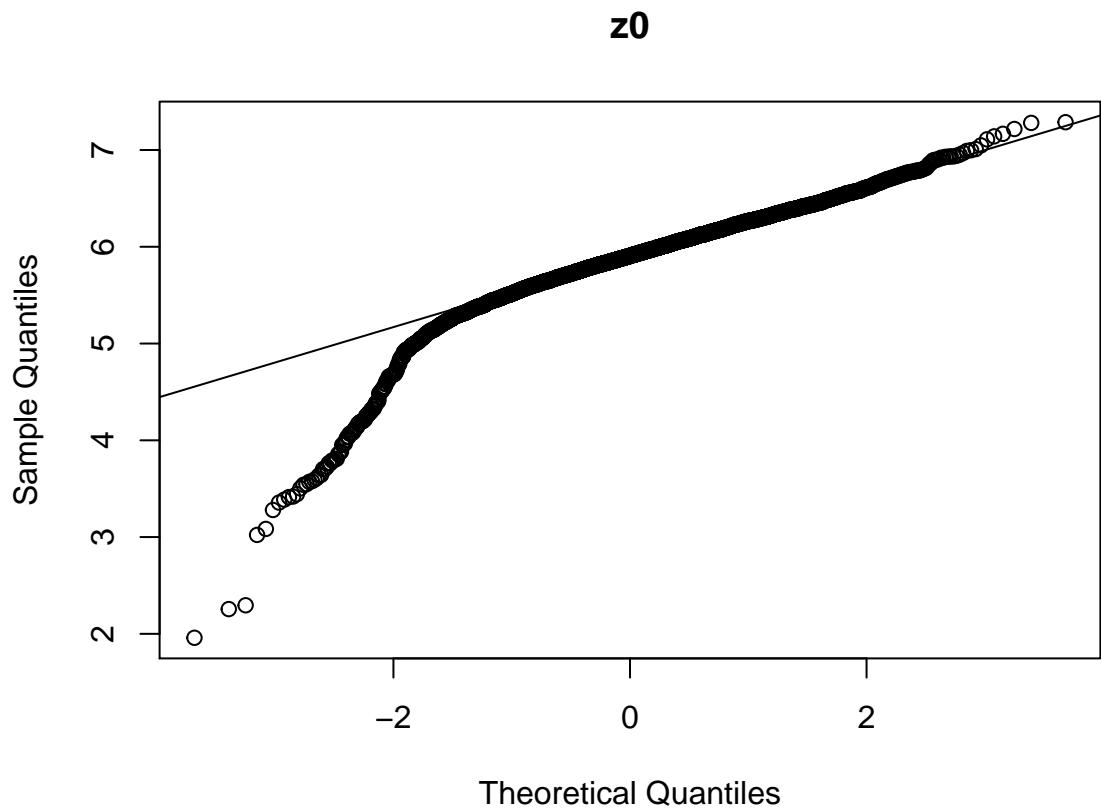




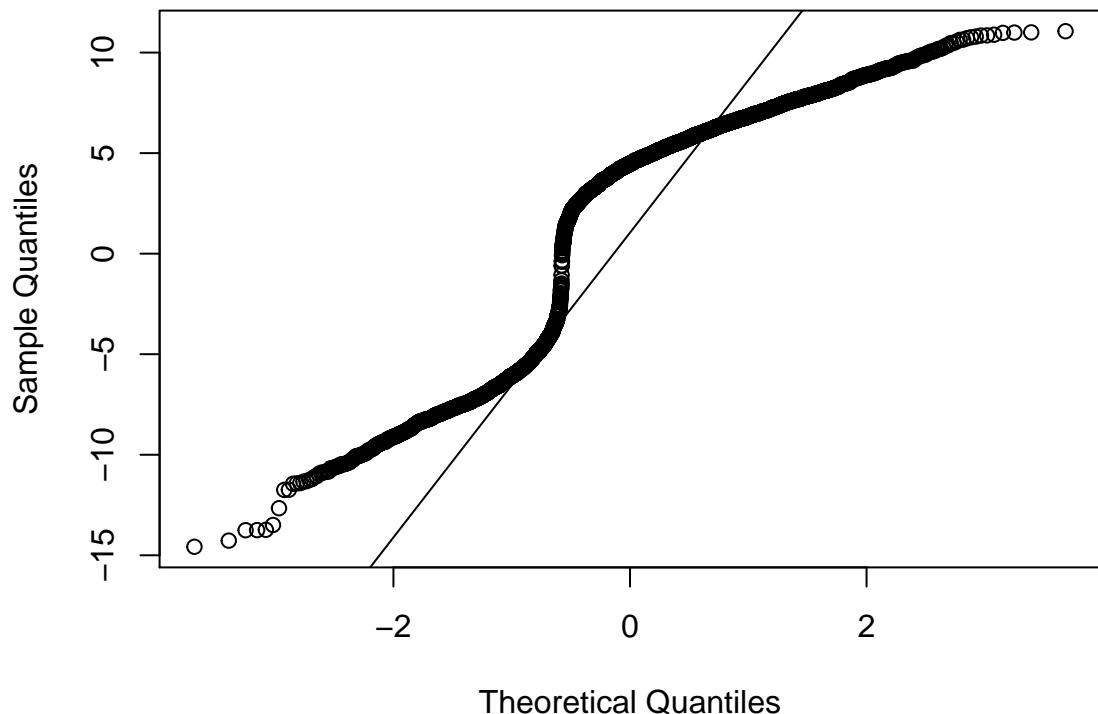




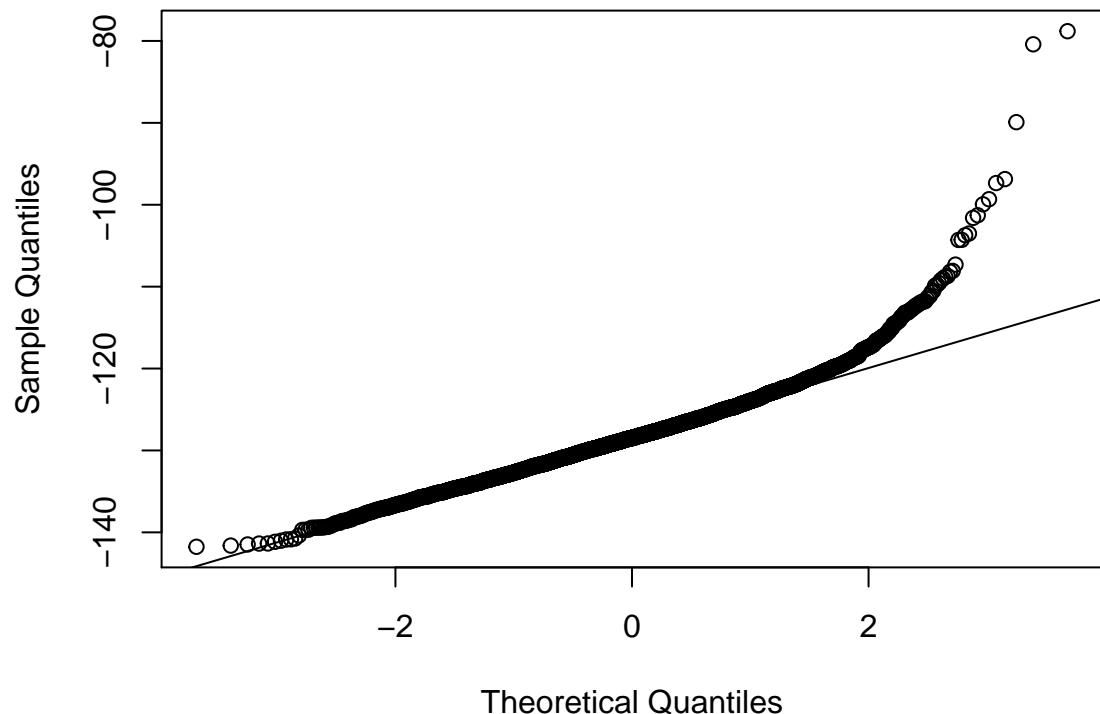




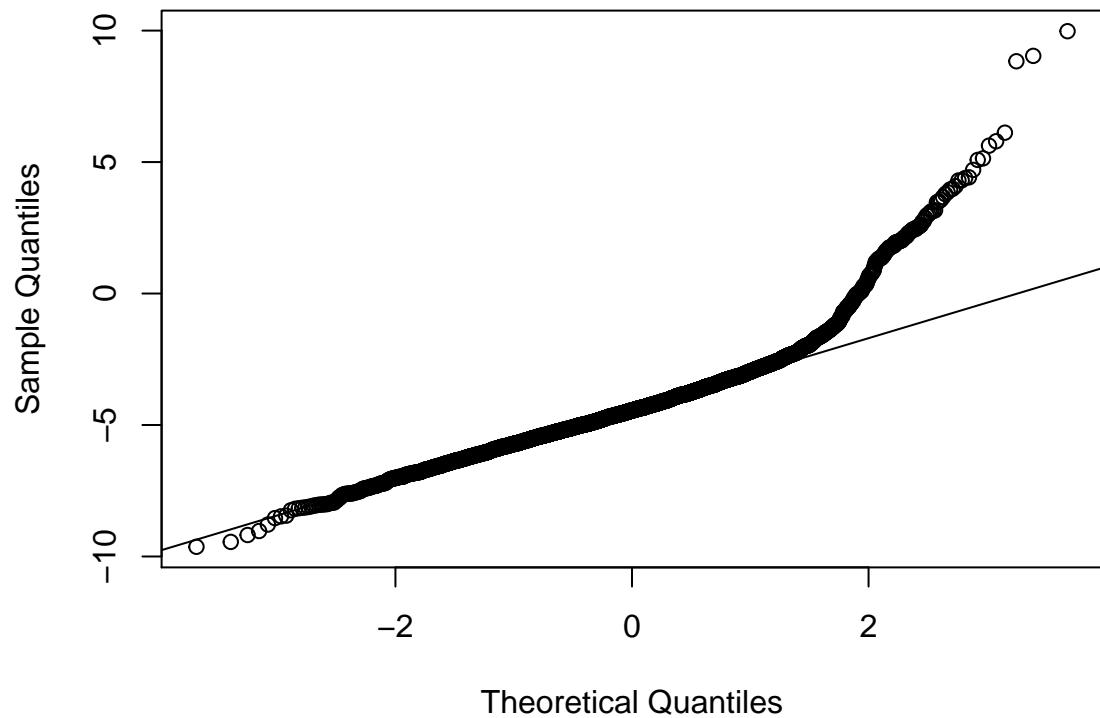
**vx0**



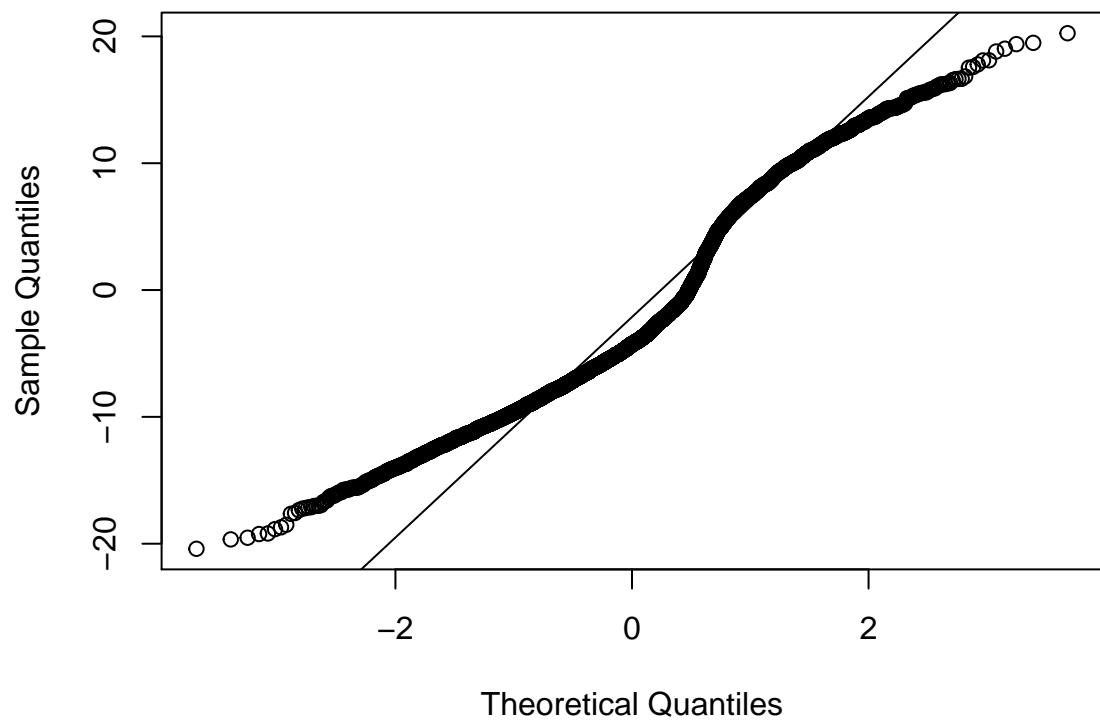
**vy0**



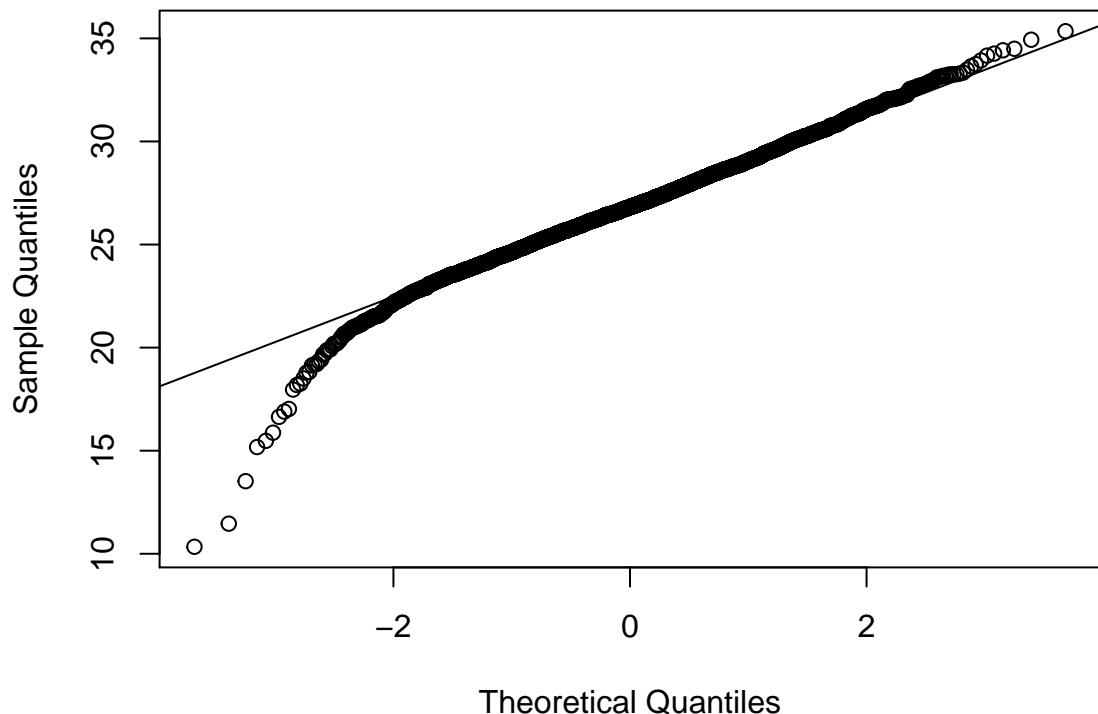
**vz0**

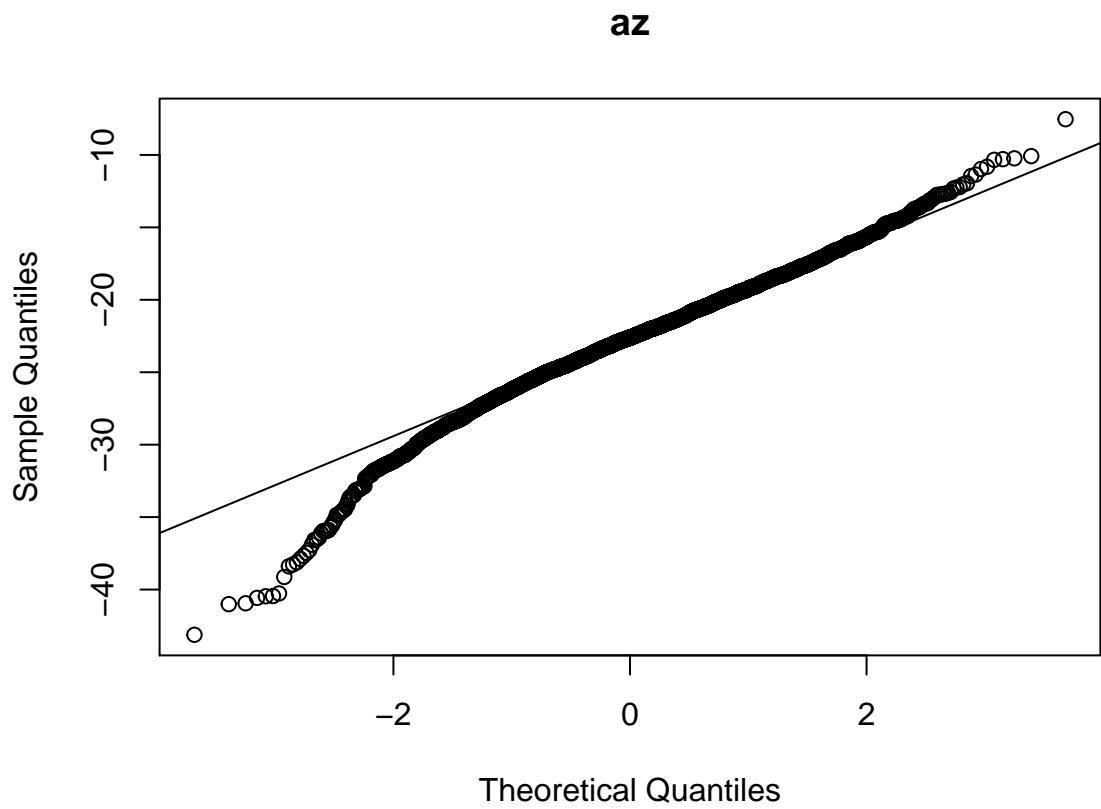


**ax**

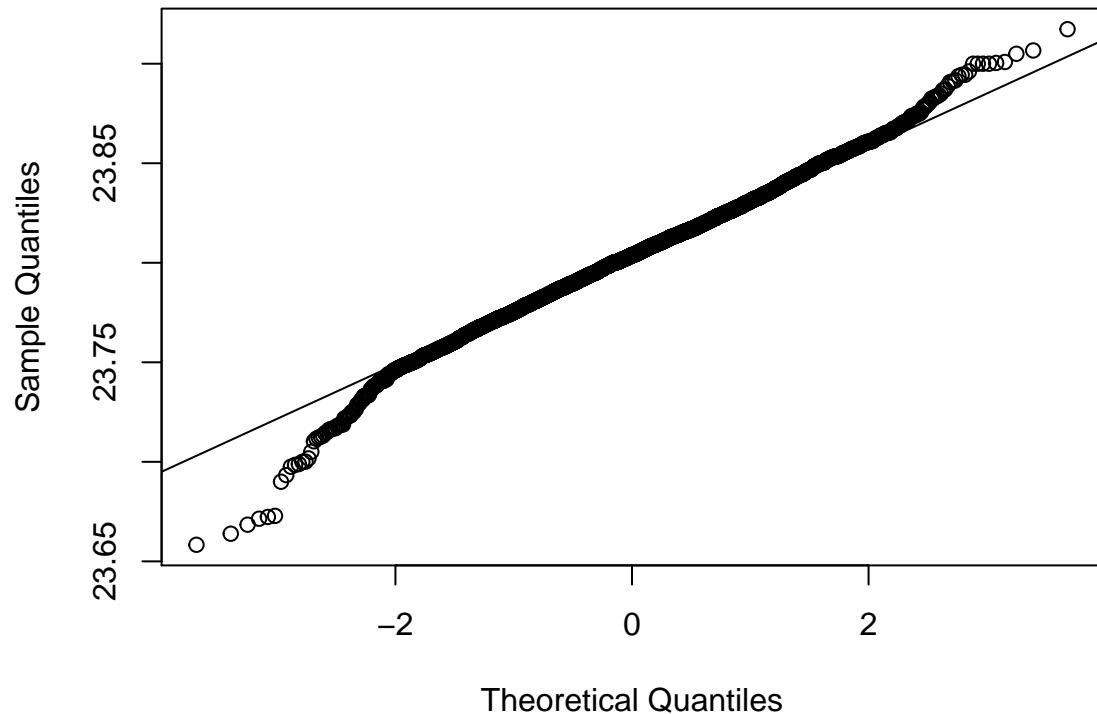


**ay**

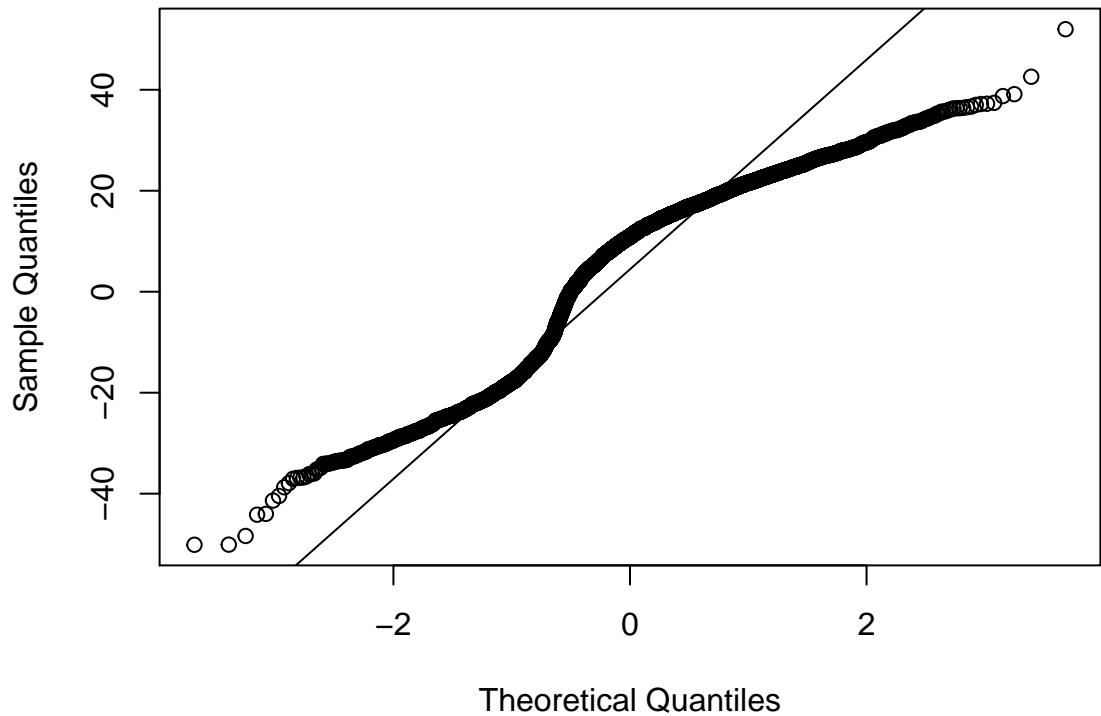




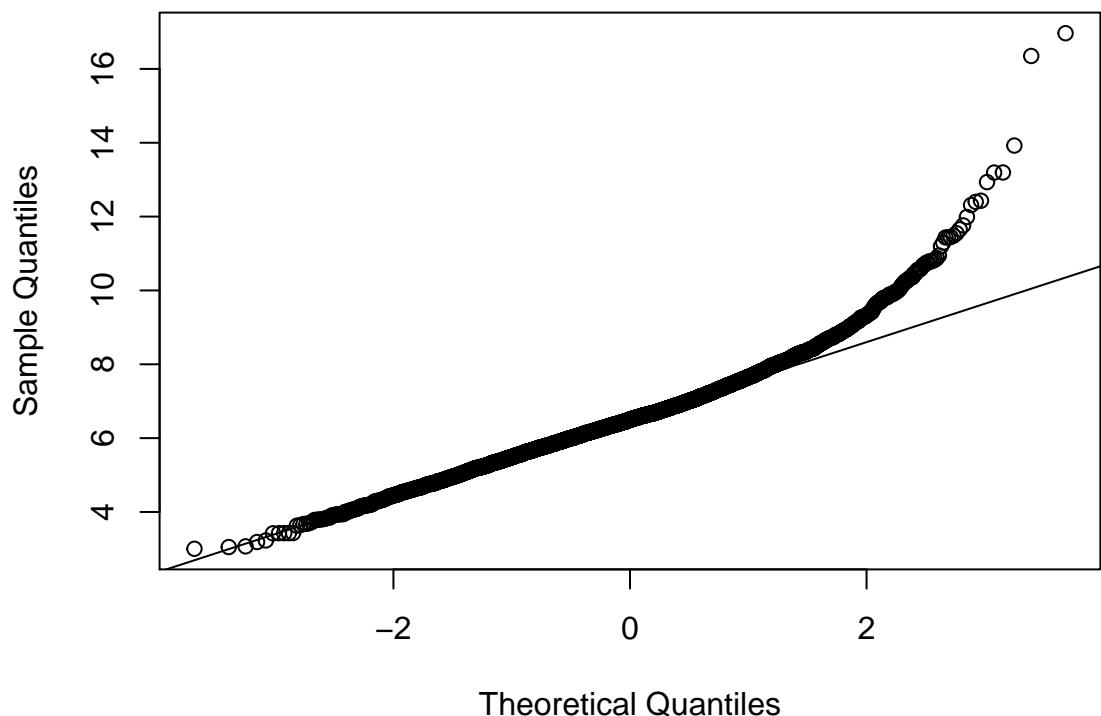
**break\_y**

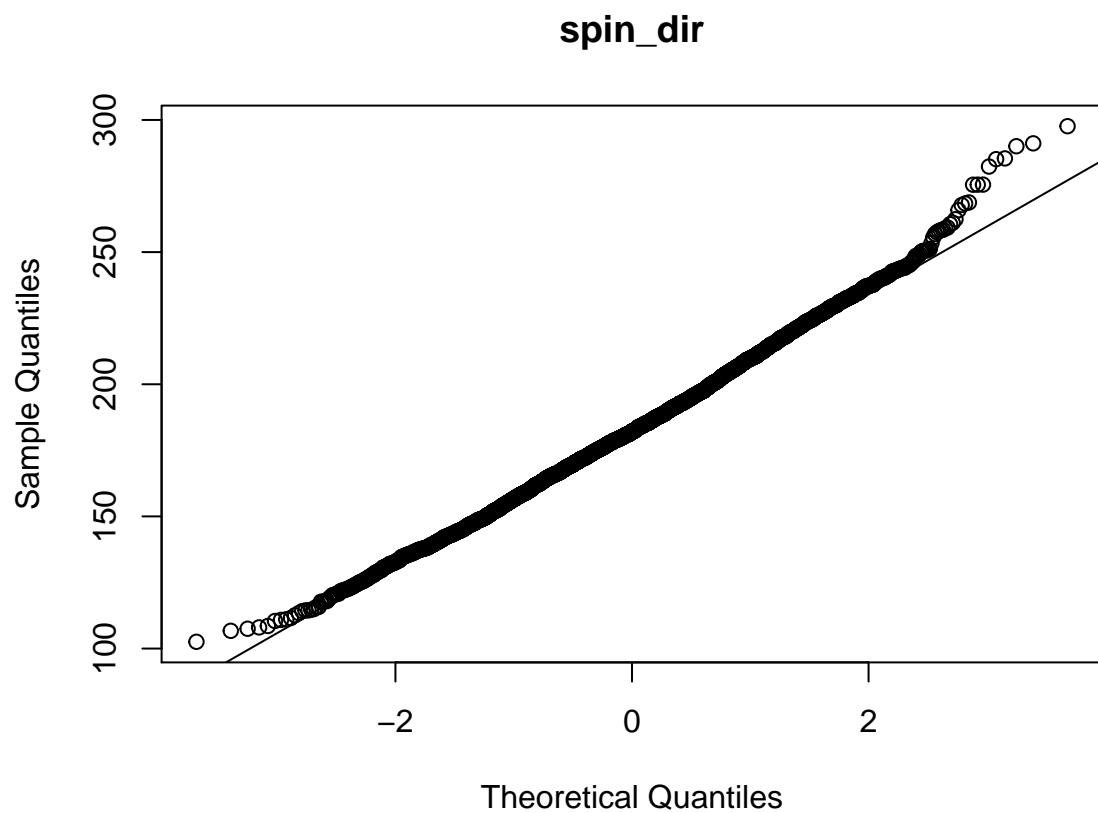


### **break\_angle**

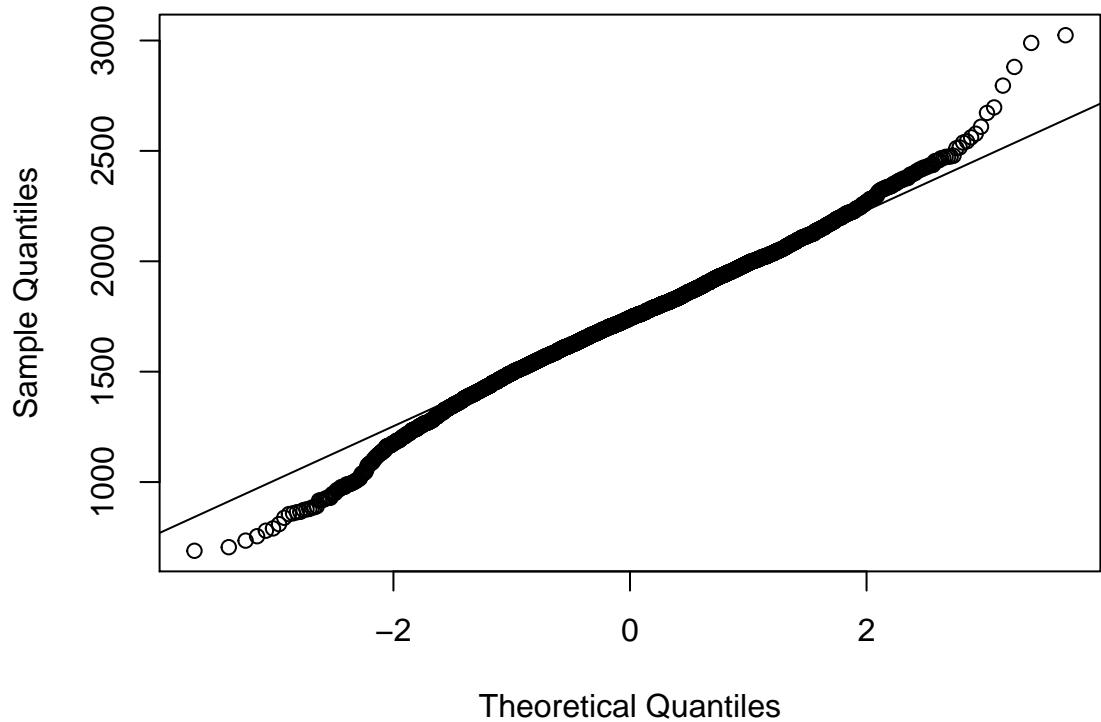


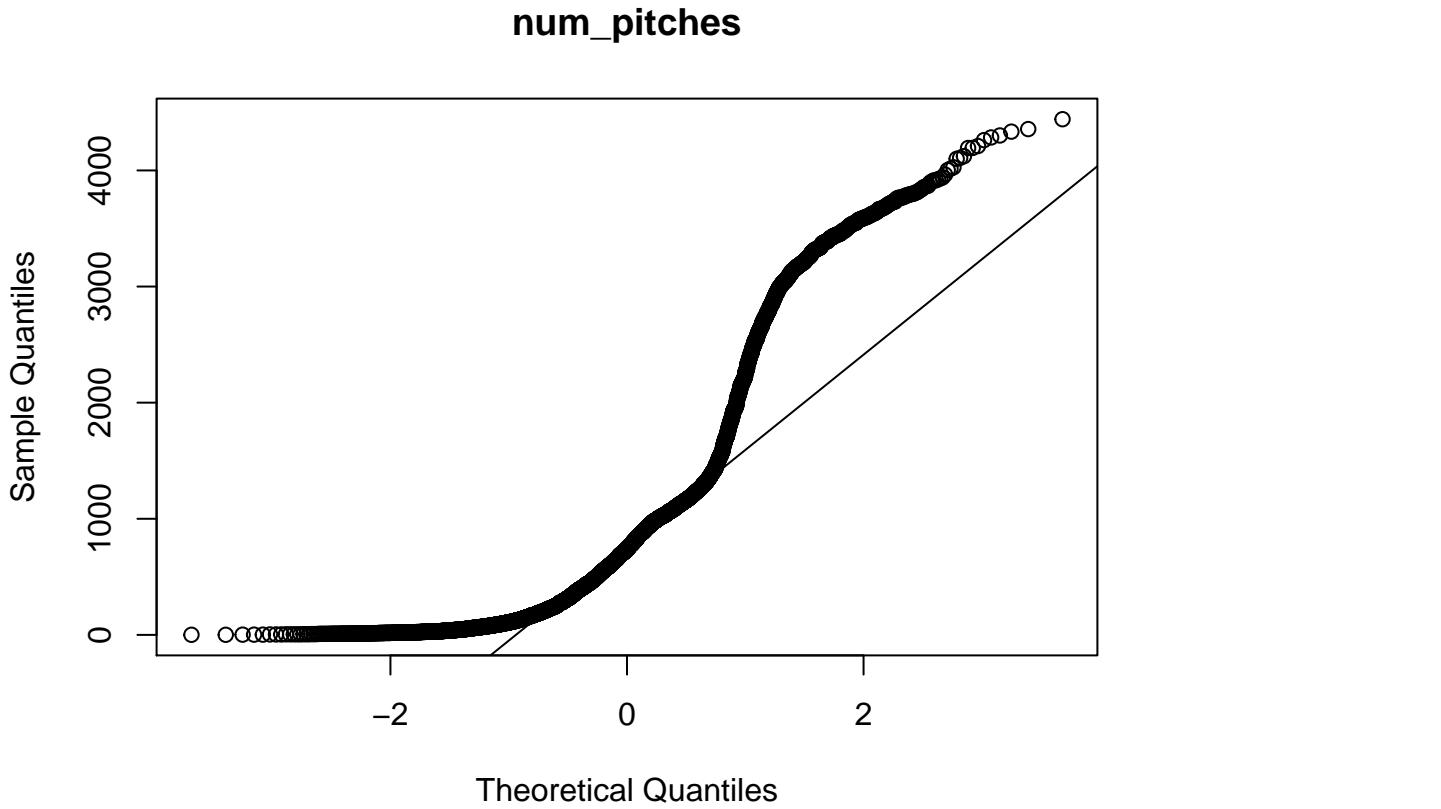
### **break\_length**





**spin\_rate**





```
#outliers <- mvOutlier(numeric_dataset, qqplot = TRUE, method = "quan");

train <- createDataPartition(pitches_dl_dataset$OnDL, p=0.65, list=FALSE);
training <- pitches_dl_dataset[train,];
testing <- pitches_dl_dataset[-train,];

selected_variables <- lowCorrelatedCols; #c('x', 'start_speed', 'y0', 'break_y', 'spin_dir', 'spin_rate');

selected_i <- which(colnames(pitches_dl_dataset) %in% selected_variables);

formula_text <- paste(names(pitches_dl_dataset)[ncol(pitches_dl_dataset)], "~",
                      paste(names(pitches_dl_dataset)[selected_i], collapse="+"));
formula <- as.formula(formula_text);

mod_fit <- train(formula, data=training, method="glm", family="binomial");

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
summary(mod_fit);

##
## Call:
## NULL
##
## Deviance Residuals:
##      Min      1Q   Median      3Q      Max
## -1.5207 -0.7147 -0.5911 -0.4217  2.3907
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.354e+07  6.325e+06 -2.141   0.0323 *
## y            2.761e-02  5.086e-03  5.429  5.68e-08 ***
## sz_bot       7.517e-01  9.399e-01  0.800   0.4239
## pfx_z        1.932e-02  2.780e-02  0.695   0.4872
## pz           3.167e-01  2.982e-01  1.062   0.2883
## x0          -7.111e-02  3.237e-02 -2.197   0.0280 *
## y0           2.708e+05  1.265e+05  2.141   0.0323 *
## z0           1.041e-01  1.251e-01  0.832   0.4053
## break_y     -5.329e-01  2.008e+00 -0.265   0.7907
## spin_dir    -4.698e-03  2.168e-03 -2.166   0.0303 *
## spin_rate   -2.820e-05  2.287e-04 -0.123   0.9019
## num_pitches 4.541e-04  4.221e-05 10.756 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3041.7 on 2813 degrees of freedom
## Residual deviance: 2849.4 on 2802 degrees of freedom
## AIC: 2873.4
##
## Number of Fisher Scoring iterations: 7
coef(mod_fit$finalModel);

## (Intercept)          y      sz_bot      pfx_z         pz
## -1.354046e+07  2.760980e-02  7.516726e-01  1.931639e-02  3.166523e-01
## x0              y0          z0      break_y      spin_dir
## -7.110976e-02  2.708093e+05  1.041405e-01 -5.328771e-01 -4.697587e-03
## spin_rate      num_pitches
## -2.819573e-05  4.540725e-04

pred <- predict(mod_fit, newdata=testing);
confusionMatrix(data=pred, reference=testing$OnDL);

```

```

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   NO   YES
##           NO  1126  330
##           YES    38   19
##
##                 Accuracy : 0.7568
##                   95% CI : (0.7343, 0.7782)
##       No Information Rate : 0.7693
##       P-Value [Acc > NIR] : 0.8826
##
##                 Kappa : 0.0308
## McNemar's Test P-Value : <2e-16
##
##                 Sensitivity : 0.96735
##                 Specificity : 0.05444
##      Pos Pred Value : 0.77335
##      Neg Pred Value : 0.33333
##                 Prevalence : 0.76933
##                 Detection Rate : 0.74422
## Detection Prevalence : 0.96233
##      Balanced Accuracy : 0.51090
##
##      'Positive' Class : NO
##

formula <- as.formula('OnDL ~ y + x0 + spin_dir + spin_rate + num_pitches');

mod_fit <- train(formula, data=training, method="glm", family="binomial");
summary(mod_fit);

##
## Call:
## NULL
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -1.4970  -0.7144  -0.5919  -0.4511   2.2231
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.2065583  0.8340137 -5.044 4.56e-07 ***
## y            0.0215149  0.0044891  4.793 1.65e-06 ***
## x0          -0.0705535  0.0317962 -2.219  0.0265 *
## spin_dir    -0.0054245  0.0021069 -2.575  0.0100 *
## spin_rate    0.0001051  0.0001845  0.570  0.5688
## num_pitches  0.0004613  0.0000414 11.143 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3041.7  on 2813  degrees of freedom
## Residual deviance: 2863.6  on 2808  degrees of freedom

```

```

## AIC: 2875.6
##
## Number of Fisher Scoring iterations: 4
coef(mod_fit$finalModel);

## (Intercept)          y          x0      spin_dir      spin_rate
## -4.2065583002  0.0215148941 -0.0705534894 -0.0054245213  0.0001051418
## num_pitches
## 0.0004612654

pred <- predict(mod_fit, newdata=testing);
confusionMatrix(data=pred, reference=testing$OnDL);

## Confusion Matrix and Statistics
##
##             Reference
## Prediction   NO    YES
##           NO 1125   331
##           YES   39    18
##
##           Accuracy : 0.7555
##           95% CI : (0.733, 0.7769)
##   No Information Rate : 0.7693
##   P-Value [Acc > NIR] : 0.9046
##
##           Kappa : 0.0256
##   Mcnemar's Test P-Value : <2e-16
##
##           Sensitivity : 0.96649
##           Specificity : 0.05158
##   Pos Pred Value : 0.77266
##   Neg Pred Value : 0.31579
##           Prevalence : 0.76933
##           Detection Rate : 0.74356
##   Detection Prevalence : 0.96233
##           Balanced Accuracy : 0.50904
##
##           'Positive' Class : NO
##
prediction <- predict(mod_fit, newdata=pitches_dl_predict, type='prob');
prediction <- cbind(pitches_dl_predict, prediction);

prediction_df <- as.data.frame(prediction);
prediction_df <- prediction_df[order(prediction_df$YES, decreasing = TRUE),];
head(prediction_df[c('nameFirst', 'nameLast', 'num_pitches', 'YES')]);

##      nameFirst nameLast num_pitches      YES
## 2299     Corey    Kluber      3950 0.6546497
## 124      Jake    Arrieta      3599 0.6482513
## 4010     Max    Scherzer      3876 0.6435166
## 3399     Martin   Perez      3277 0.6399398
## 2307     Tom    Koehler      3216 0.6283903
## 2845     Collin  McHugh      3296 0.6208720

```

```

selected_variables <- lowCorrelatedCols; #c('x', 'start_speed', 'y0', 'break_y', 'spin_dir', 'num_pitches')

selected_i <- which(colnames(pitches_dl_dataset) %in% selected_variables);

formula_text <- paste(names(pitches_dl_dataset)[ncol(pitches_dl_dataset)], "~",
                      paste(names(pitches_dl_dataset)[selected_i], collapse="+")));
formula <- as.formula(formula_text);

mod_1 = glm(formula = formula , family=binomial(logit), data=training);

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
summary(mod_1);

## 
## Call:
## glm(formula = formula, family = binomial(logit), data = training)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.5207   -0.7147   -0.5911   -0.4217    2.3907
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.354e+07 6.325e+06 -2.141  0.0323 *
## y            2.761e-02 5.086e-03  5.429 5.68e-08 ***
## sz_bot       7.517e-01 9.399e-01  0.800  0.4239
## pfx_z        1.932e-02 2.780e-02  0.695  0.4872
## pz           3.167e-01 2.982e-01  1.062  0.2883
## x0          -7.111e-02 3.237e-02 -2.197  0.0280 *
## y0           2.708e+05 1.265e+05  2.141  0.0323 *
## z0           1.041e-01 1.251e-01  0.832  0.4053
## break_y      -5.329e-01 2.008e+00 -0.265  0.7907
## spin_dir     -4.698e-03 2.168e-03 -2.166  0.0303 *
## spin_rate    -2.820e-05 2.287e-04 -0.123  0.9019
## num_pitches  4.541e-04 4.221e-05 10.756 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3041.7 on 2813 degrees of freedom
## Residual deviance: 2849.4 on 2802 degrees of freedom
## AIC: 2873.4
##
## Number of Fisher Scoring iterations: 7

pred <- ifelse(predict(mod_1, testing, type='response') > 0.5, 'YES', 'NO')
confusionMatrix(data=pred, reference=testing$OnDL);

## Confusion Matrix and Statistics
## 
##             Reference
## Prediction    NO    YES
##           NO 1126  330

```

```

##      YES    38    19
##
##          Accuracy : 0.7568
##                95% CI : (0.7343, 0.7782)
##      No Information Rate : 0.7693
##      P-Value [Acc > NIR] : 0.8826
##
##          Kappa : 0.0308
## McNemar's Test P-Value : <2e-16
##
##          Sensitivity : 0.96735
##          Specificity : 0.05444
##      Pos Pred Value : 0.77335
##      Neg Pred Value : 0.33333
##          Prevalence : 0.76933
##      Detection Rate : 0.74422
##      Detection Prevalence : 0.96233
##      Balanced Accuracy : 0.51090
##
##      'Positive' Class : NO
##

formula_text <- paste(names(training)[ncol(training)], "~1");
formula <- as.formula(formula_text);

mod_nothing = glm(formula = formula , family=binomial(logit), data=training);

summary(mod_nothing);

##
## Call:
## glm(formula = formula, family = binomial(logit), data = training)
##
## Deviance Residuals:
##      Min        1Q     Median        3Q       Max
## -0.7248   -0.7248   -0.7248   -0.7248    1.7120
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.20274    0.04473 -26.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 3041.7  on 2813  degrees of freedom
## Residual deviance: 3041.7  on 2813  degrees of freedom
## AIC: 3043.7
##
## Number of Fisher Scoring iterations: 4
#backward <- step(glm.out);

#summary(backwards);

```

```
#forward <- step(glm.out, direction = "forward");  
#summary(forward);
```