Wisdom of Crowds at Threadless

Juho Salminen

22 Apr 2015

# Summary of the dataset

## Dataset consists of 15592 observations of 10 variables.  
## 194 rows with missing values have been removed.

Names of the variables and a sample of data:

## [1] "approved\_date" "design" "user" "avg.score"   
## [5] "score" "fives" "ones" "printed"   
## [9] "challenge" "printed.binary"

## approved\_date design user avg.score score fives ones  
## 1 2012-07-24 design\_27687 user\_1 3.88 26 10 1  
## 2 2012-07-24 design\_20167 user\_501 4.09 23 10 0  
## 3 2012-07-30 design\_16016 user\_720 2.67 9 1 1  
## 4 2012-07-30 design\_27435 user\_1522 3.00 10 1 0  
## 5 2012-07-30 design\_27716 user\_12033 3.73 11 4 0  
## 6 2012-07-30 design\_24325 user\_11546 2.91 11 1 1  
## printed challenge printed.binary  
## 1 Not printed threadless 0  
## 2 Not printed threadless 0  
## 3 Not printed threadless 0  
## 4 Not printed threadless 0  
## 5 Not printed threadless 0  
## 6 Not printed threadless 0

## Descriptive statistics

## avg.score score fives ones   
## Min. :1.550 Min. : 7.0 Min. :-115.00 Min. : 0.00   
## 1st Qu.:2.470 1st Qu.: 222.0 1st Qu.: 23.00 1st Qu.: 44.00   
## Median :2.750 Median : 294.0 Median : 39.00 Median : 59.00   
## Mean :2.754 Mean : 339.8 Mean : 54.86 Mean : 68.78   
## 3rd Qu.:3.020 3rd Qu.: 406.0 3rd Qu.: 66.00 3rd Qu.: 82.00   
## Max. :4.690 Max. :4961.0 Max. :4175.00 Max. :722.00   
## printed   
## Not printed:15387   
## Printed : 205   
##   
##   
##   
##

## avg.score score fives ones printed  
## median 2.750000000 2.940000e+02 39.0000000 59.0000000 NA  
## mean 2.753923807 3.398473e+02 54.8601847 68.7793099 NA  
## SE.mean 0.003161038 1.517120e+00 0.6384537 0.3123565 NA  
## CI.mean.0.95 0.006196002 2.973731e+00 1.2514434 0.6122551 NA  
## var 0.155797777 3.588737e+04 6355.6594960 1521.2583930 NA  
## std.dev 0.394712271 1.894396e+02 79.7223902 39.0033126 NA  
## coef.var 0.143327230 5.574257e-01 1.4531922 0.5670791 NA

## approved\_date design user avg.score score fives ones  
## 25681 2013-04-29 design\_21031 user\_3060 2.92 231 -115 32  
## printed challenge printed.binary  
## 25681 Not printed threadless 0

## 1.31 % of designs have been printed.

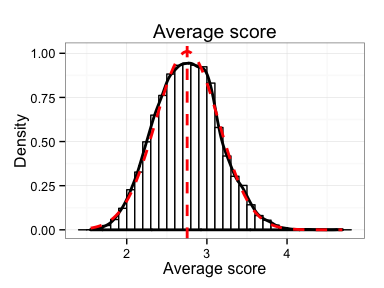
Average score of designs varies between 1.55 and 4.69, close to smallest and largest possible average scores 1 and 5. Numbers of scores, fives and ones are not limited, and indeed some designs have had thousands of people scoring them. Fives have an outlier: for some reason one of the values is negative. Only a small minority of designs in the data have been printed.

## Descriptive statistics by print status

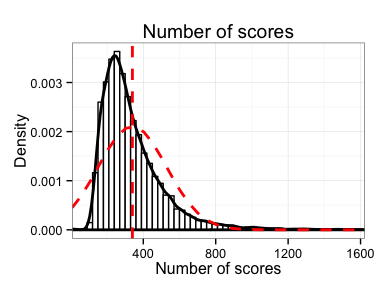
## threadless$printed: Not printed  
## avg.score score fives ones   
## Min. :1.550 Min. : 7.0 Min. :-115.00 Min. : 0.00   
## 1st Qu.:2.470 1st Qu.: 221.0 1st Qu.: 23.00 1st Qu.: 44.00   
## Median :2.750 Median : 293.0 Median : 39.00 Median : 59.00   
## Mean :2.748 Mean : 338.5 Mean : 54.03 Mean : 68.93   
## 3rd Qu.:3.010 3rd Qu.: 404.0 3rd Qu.: 65.00 3rd Qu.: 82.00   
## Max. :4.690 Max. :4961.0 Max. :4175.00 Max. :722.00   
## --------------------------------------------------------   
## threadless$printed: Printed  
## avg.score score fives ones   
## Min. :2.010 Min. : 104.0 Min. : 6.0 Min. : 15.00   
## 1st Qu.:2.880 1st Qu.: 268.0 1st Qu.: 46.0 1st Qu.: 38.00   
## Median :3.240 Median : 380.0 Median : 89.0 Median : 53.00   
## Mean :3.179 Mean : 439.7 Mean : 116.9 Mean : 57.62   
## 3rd Qu.:3.500 3rd Qu.: 549.0 3rd Qu.: 151.0 3rd Qu.: 71.00   
## Max. :4.280 Max. :1720.0 Max. :1153.0 Max. :161.00

Printed designs tend to have higher average scores: mean 3.2 for printed desings vs. 2.75 for designs that have not been printed. Numbers of scores and fives are also a little higher compared to designs that have not been printed.

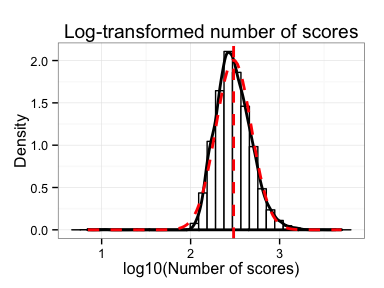
# Distributions of single variables



The distribution of average scores (solid black line) appears to follow closely normal distribution with same mean and standard deviations (dashed red line).

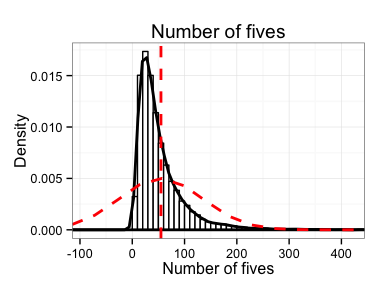


Distribution of number of scores is right-skewed and does not resemble normal distribution (dashed red line). Median is a better summary than mean for this distribution.

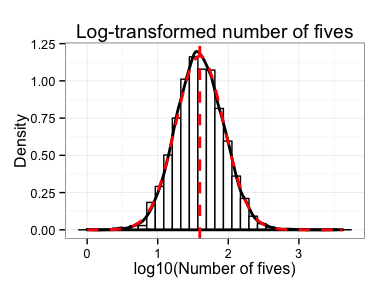


Log transformation fixes the skewness and gets the distribution much closer to the normal distribution.

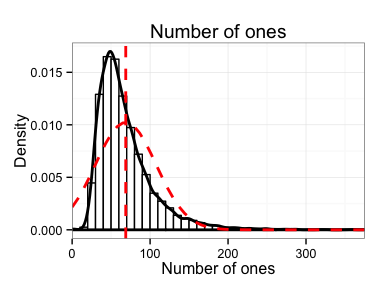
## Fives



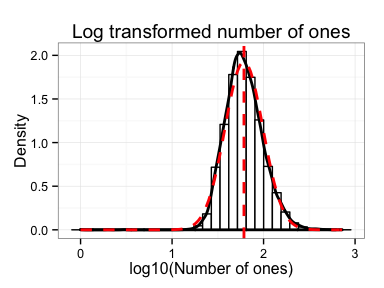
Distribution of number of fives is also right-skewed and does not follow normal distribution.



Log transformed data closely resembles a normal distribution

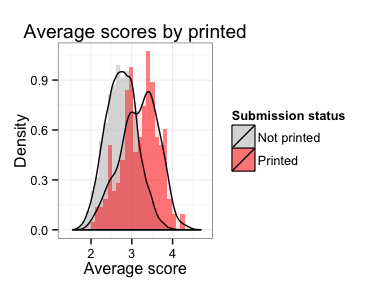


Similar right-skewed distribution again as with the scores and fives.

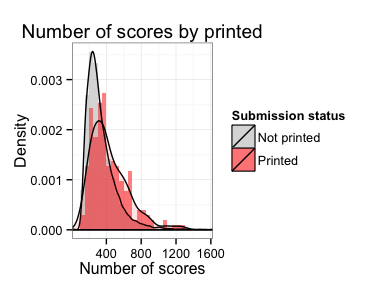


Log transformed data again resembles normal distribution. If scores, fives or ones are used in further analyses, it is probably best to use them in log transformed formats.

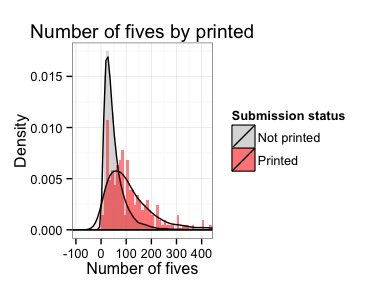
# Relationships between variables



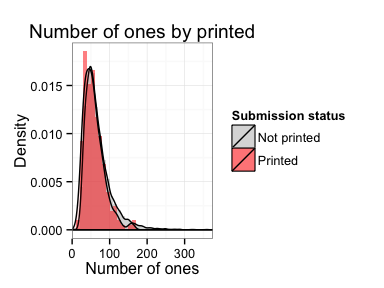
Printed designs have a tendency to have higher average scores than designs that have not been printed, but there's quite a lot of overlap.



Distribution of number of scores is shifted slightly right for printed designs. The difference is small.

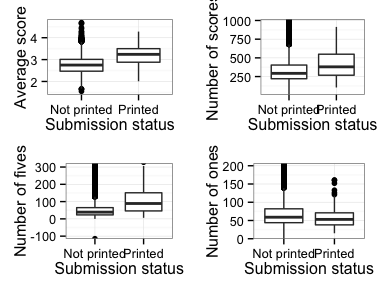


Distribution of fives for printed designs has fatter right tail than the distribution for designs that have not been printed. If design gathers more than 150 fives it appears to have good changes of getting printed.



Distributions for numbers of ones are almost identical for printed and not printed designs.

## Boxplots



Based on the boxplots average score looks like the best predictor of designs getting printed. Number of fives is probably the second best, followed by number of scores. Number of ones appears useless in predicting the print status.

# Data transformations

# Remove rows where variable to be log transformed is 0  
remove <- threadless$score <= 0 | threadless$fives <= 0 | threadless$ones <= 0  
threadless <- threadless[!remove, ]  
cat("Removed", sum(remove), "observations.")

## Removed 5 observations.

threadless[remove, c(4:8)]

## avg.score score fives ones printed  
## 3 2.67 9 1 1 Not printed  
## 8 3.67 33 15 3 Not printed  
## 10 3.00 11 3 2 Not printed  
## 2923 2.43 129 10 31 Not printed  
## 25689 2.52 247 34 80 Not printed

# Separate number of fives from the average score  
threadless$total <- threadless$avg.score \* threadless$score  
threadless$avg.wo5 <- (threadless$total - threadless$fives \* 5) /   
 (threadless$score - threadless$fives)  
  
# Log transform the data as necessary and normalize  
threadless$avg.score.norm <- scale(threadless$avg.score)  
threadless$score.norm <- scale(log10(threadless$score))  
threadless$fives.norm <- scale(log10(threadless$fives))  
threadless$ones.norm <- scale(log10(threadless$ones))  
threadless$avg.wo5.norm <- scale(threadless$avg.wo5)

# Logistic regression analysis

##   
## Call:  
## glm(formula = printed ~ avg.score, family = binomial(), data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2775 -0.1740 -0.1223 -0.0847 3.6909   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -12.0920 0.6688 -18.08 <2e-16 \*\*\*  
## avg.score 2.6278 0.2103 12.50 <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: 1523.8 on 10909 degrees of freedom  
## Residual deviance: 1363.1 on 10908 degrees of freedom  
## AIC: 1367.1  
##   
## Number of Fisher Scoring iterations: 8

The model estimates the probabilities of designs being printed based on the average score they have gathererd. It is statistically significant. The probability of getting this kind of data by change if there was no statistical effect between average score and design getting printed is practically zero.

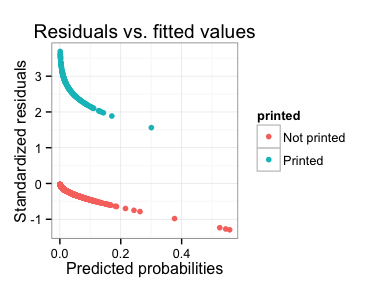
## modelChi 160.724   
## chidf 1   
## chisq.prob 0   
## Pseudo R^2 for logistic regression  
## Hosmer and Lemeshow R^2 0.105   
## Cox and Snell R^2 0.015   
## Nagelkerke R^2 0.112   
## Odds ratios:  
## (Intercept) avg.score   
## 0.000 13.843   
## Confidence intervals:

## Waiting for profiling to be done...

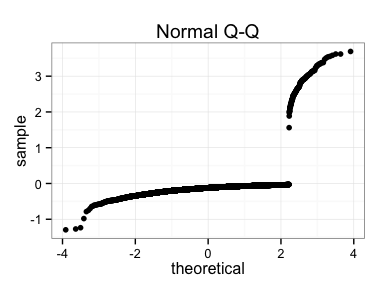
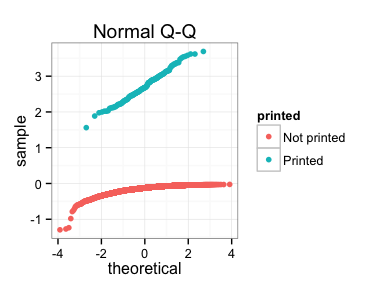
## 0.5 % 99.5 %  
## (Intercept) 0.000 0.000  
## avg.score 8.093 23.965

Chi-squared test shows the model fits the data significantly better than random change.Pseudo R^2 statistics still suggest that the effect is not very large. The average scores alone cannot explain which designs get printed. The odds ratio of average score is 15.7 with 99 % confidence interval from 9.1 to 27.2. Getting 1 unit better average score increases the probability of design getting printed by 9 to 27 fold.

## Model diagnostics



Designs that have not been printed have relatively low residuals. On the other hand printed designs have problematically large residuals. The model does not work well with the designs that do get printed.

Q-Q plot does not look good either, presumably because of the model's poor performance with printed designs.

## 139 observations have residuals larger than 2 standard deviations.  
## 650 observations have leverage more than 2 times larger than the average.

Observations with large residuals:

##   
## Not printed Printed  
## FALSE 10767 4  
## TRUE 0 139

All cases with problematically large residuals are printed designs.

Observations with large leverage:

##   
## Not printed Printed  
## FALSE 10168 92  
## TRUE 599 51

Observations with large leverage on the model are more equally distributed among printed and not printed designs.

## train$printed[train$large.leverage]: Not printed  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.390 3.440 3.500 3.553 3.610 4.690   
## --------------------------------------------------------   
## train$printed[train$large.leverage]: Printed  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.390 3.470 3.610 3.635 3.760 4.280

## train$printed[!train$large.leverage]: Not printed  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.550 2.450 2.710 2.701 2.970 3.380   
## --------------------------------------------------------   
## train$printed[!train$large.leverage]: Printed  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.010 2.738 2.990 2.924 3.230 3.380

sum(train$avg.score >= 3.39 & train$printed == "Not printed")

## [1] 599

sum(train$avg.score >= 3.39 & train$printed == "Printed")

## [1] 51

Both printed and not printed designs with high scores have large leverage. Perhaps this has something to do with the fact that most of the designs do not get printed.

## Improvements to the model

##   
## CORRELATIONS  
## ============  
## - correlation type: pearson   
## - correlations shown only when both variables are numeric  
##   
## avg.score score fives ones ones.norm  
## avg.score . 0.360 0.518 -0.347 .  
## score 0.360 . 0.717 0.609 .  
## fives 0.518 0.717 . 0.185 .  
## ones -0.347 0.609 0.185 . .  
## ones.norm . . . . .

Correlations between variables do not look so large that multicollinearity would be a problem.

# More complex linear regression models. Using normalized and   
# log transformed variables  
model.2 <- glm(printed ~ avg.score.norm, data = train, family = 'binomial')  
model.3 <- update(model.2, .~. + fives.norm)  
model.4 <- update(model.3, .~. + score.norm)  
summary(model.2)

##   
## Call:  
## glm(formula = printed ~ avg.score.norm, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2775 -0.1740 -0.1223 -0.0847 3.6909   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.85564 0.11998 -40.47 <2e-16 \*\*\*  
## avg.score.norm 1.03658 0.08295 12.50 <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: 1523.8 on 10909 degrees of freedom  
## Residual deviance: 1363.1 on 10908 degrees of freedom  
## AIC: 1367.1  
##   
## Number of Fisher Scoring iterations: 8

summary(model.3)

##   
## Call:  
## glm(formula = printed ~ avg.score.norm + fives.norm, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2789 -0.1740 -0.1223 -0.0848 3.6907   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.855476 0.120218 -40.389 < 2e-16 \*\*\*  
## avg.score.norm 1.033717 0.160899 6.425 1.32e-10 \*\*\*  
## fives.norm 0.003182 0.153233 0.021 0.983   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1523.8 on 10909 degrees of freedom  
## Residual deviance: 1363.1 on 10907 degrees of freedom  
## AIC: 1369.1  
##   
## Number of Fisher Scoring iterations: 8

summary(model.4)

##   
## Call:  
## glm(formula = printed ~ avg.score.norm + fives.norm + score.norm,   
## family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2567 -0.1744 -0.1224 -0.0843 3.6926   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.86143 0.12171 -39.942 < 2e-16 \*\*\*  
## avg.score.norm 0.96767 0.23583 4.103 4.07e-05 \*\*\*  
## fives.norm 0.13638 0.38157 0.357 0.721   
## score.norm -0.08666 0.22764 -0.381 0.703   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1523.8 on 10909 degrees of freedom  
## Residual deviance: 1362.9 on 10906 degrees of freedom  
## AIC: 1370.9  
##   
## Number of Fisher Scoring iterations: 8

anova(model.2, model.3, model.4, test = "Chisq")

## Analysis of Deviance Table  
##   
## Model 1: printed ~ avg.score.norm  
## Model 2: printed ~ avg.score.norm + fives.norm  
## Model 3: printed ~ avg.score.norm + fives.norm + score.norm  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)  
## 1 10908 1363.1   
## 2 10907 1363.1 1 0.000431 0.9834  
## 3 10906 1362.9 1 0.144757 0.7036

After using average score to predict print status adding more variables does not improve the model. Neither number of fives or number of scores is statistically significant. Comparing the models using anova further confirms the lack of improvement.

##   
## Call:  
## glm(formula = printed ~ avg.wo5.norm, family = "binomial", data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5528 -0.1785 -0.1272 -0.0882 3.7182   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.80546 0.11988 -40.08 <2e-16 \*\*\*  
## avg.wo5.norm 1.01651 0.09198 11.05 <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: 1523.8 on 10909 degrees of freedom  
## Residual deviance: 1390.8 on 10908 degrees of freedom  
## AIC: 1394.8  
##   
## Number of Fisher Scoring iterations: 8

##   
## Call:  
## glm(formula = printed ~ avg.wo5.norm + fives.norm, family = "binomial",   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.9492 -0.1768 -0.1220 -0.0838 3.7383   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.8595 0.1228 -39.584 < 2e-16 \*\*\*  
## avg.wo5.norm 0.6309 0.1205 5.235 1.65e-07 \*\*\*  
## fives.norm 0.5219 0.1158 4.508 6.53e-06 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1523.8 on 10909 degrees of freedom  
## Residual deviance: 1372.7 on 10907 degrees of freedom  
## AIC: 1378.7  
##   
## Number of Fisher Scoring iterations: 8

## Analysis of Deviance Table  
##   
## Model 1: printed ~ avg.wo5.norm  
## Model 2: printed ~ avg.wo5.norm + fives.norm  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 10908 1390.8   
## 2 10907 1372.7 1 18.113 2.082e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Here the logistic regression is first performed with average score with fives removed and then number of fives is added as a variable. This time the addition of number of fives improves the model.

print\_stats(model.1)

## modelChi 160.724   
## chidf 1   
## chisq.prob 0   
## Pseudo R^2 for logistic regression  
## Hosmer and Lemeshow R^2 0.105   
## Cox and Snell R^2 0.015   
## Nagelkerke R^2 0.112   
## Odds ratios:  
## (Intercept) avg.score   
## 0.000 13.843   
## Confidence intervals:

## Waiting for profiling to be done...

## 0.5 % 99.5 %  
## (Intercept) 0.000 0.000  
## avg.score 8.093 23.965

print\_stats(model.7)

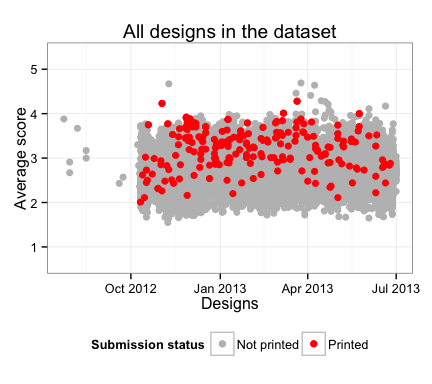
## modelChi 151.1519   
## chidf 2   
## chisq.prob 0   
## Pseudo R^2 for logistic regression  
## Hosmer and Lemeshow R^2 0.099   
## Cox and Snell R^2 0.014   
## Nagelkerke R^2 0.106   
## Odds ratios:  
## (Intercept) avg.wo5.norm fives.norm   
## 0.008 1.879 1.685   
## Confidence intervals:

## Waiting for profiling to be done...

## 0.5 % 99.5 %  
## (Intercept) 0.006 0.010  
## avg.wo5.norm 1.397 2.595  
## fives.norm 1.236 2.241

When compared to original model that used only average score, the original model has slightly higher pseudo R^2 values. Performance of the original model on the training set could not be improved by adding variables to the model.

As a conclusion, using average score to predict print status of designs provides the logistic regression model with the best fit.



This graph shows that although there is a tendency for designs with higher average score to get printed more often, it is not possible to categorize them to printed and not printed designs based on the average score. The crowd can predict, but it cannot categorize.

# Evaluating and testing the model

To validate the model the predicted probabilities between training and test sets are compared.

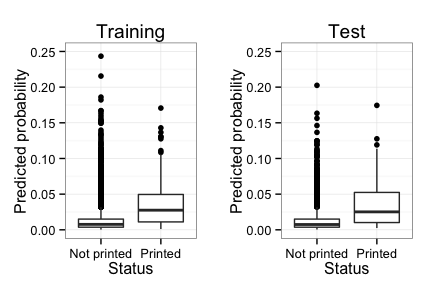
test$predicted.probabilities <- predict(model.1, newdata = test, type = "response")  
  
mse.train <- mean((train$predicted.probabilities - train$printed.binary)^2)  
mse.test <- mean((test$predicted.probabilities - test$printed.binary)^2)

Mean squared error for the training set is 0.01262 and MSE for the test set is 0.0126528. The difference is minimal, which indicates the model fits the training and tests sets equally well.

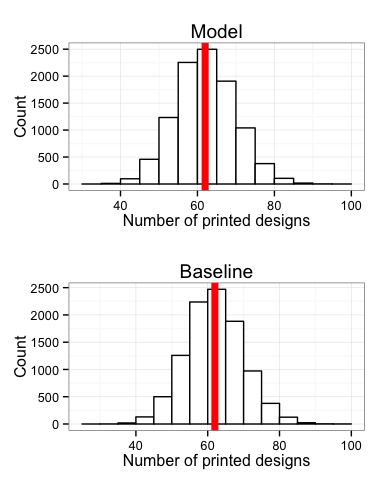
p1 <- ggplot(train) +   
 geom\_boxplot(aes(printed, predicted.probabilities)) +   
 ylim(0, 0.25) +   
 ylab("Predicted probability") +   
 xlab("Status") +   
 ggtitle("Training")  
  
p2 <- ggplot(test) +   
 geom\_boxplot(aes(printed, predicted.probabilities)) +   
 ylim(0, 0.25) +   
 ylab("Predicted probability") +   
 xlab("Status") +   
 ggtitle("Test")  
grid.arrange(p1, p2, nrow = 1)

## Warning: Removed 6 rows containing non-finite values (stat\_boxplot).

## Warning: Removed 4 rows containing non-finite values (stat\_boxplot).



The printed designs in the test set tend to have higher predicted probabilities than designs that have not been printed.

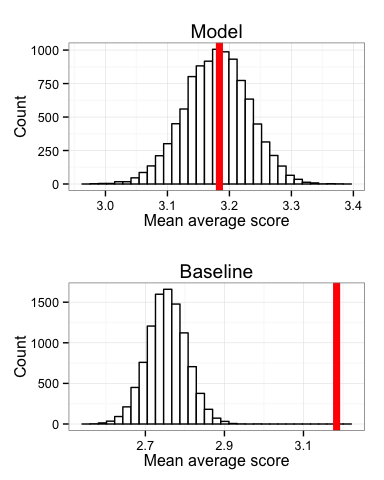
Next a simple decision making simulation is used to further validate the model. It is assumed the decision makers choose designs to be printed according to probabilities predicted by the model. Decision making on the test set is simulated 10 000 times by assigning each design a random number drawn from a uniform distrtibution. If the number is smaller or equal to predicted probability, the design is printed. Number of printed designs and mean average score and standard deviation of printed designs is stored and compared to the actual observed values and results of baseline decision making simulation. In baseline simulation each design is given equal probability of getting printed based on the probability of randomly selected design being printed in the training set (number of printed designs in training set / number of design in training set). 

Both model and baseline simulation tend to produce the similar numbers of printed designs as actually observed, which indicates the ratio of designs chosen to be printed is similar in training and test sets.

## Ratio of printed designs in training set is 0.0131   
## Ratio of printed designs in test set is 0.0133

This is indeed the case.

## stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.  
## stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.

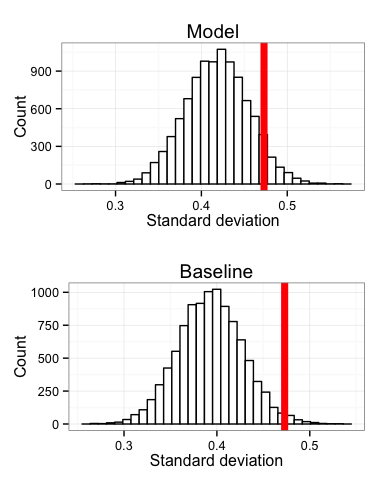


Mean average score of printed designs in model simulation are centered around the actually observed mean average score. Baseline simulation never gets the correct value. Model is thus much better fit to the data than the baseline.

## stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.

## Warning: position\_stack requires constant width: output may be incorrect

## stat\_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.



Regarding the standard deviation of average scores of printed designs the model simulation fares slightly better than the baseline simulation. It appears that in reality there is more variation in scores of printed designs than the decision making simulation typically generates.