ysisResults

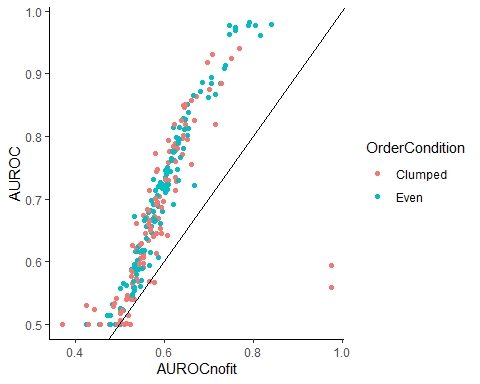
Jenny C. A. Read

21/01/2021

# Data exploration with graphs

## Overall performance as measured by no-fit AUROC

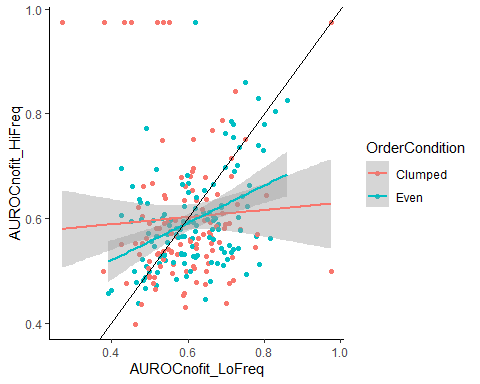
Given that the estimates of R and d’ from the fit are very noisy (have wide confidence intervals), I thought it would be worth looking at a simpler estimate of overall performance. Here, I compute d’ = erfinv(pHit) - erfinv(pFA), using the hit rate and false alarm rate based on the observer’s classifications without using the confidence judgments at all, and compute the area under the AUROC with this d’ and R=0. (If this d’ came out as >4, I truncated it at 4 to avoid numerical problems.) This provides a simple measure of overall performance which does not depend on the fitting. Since this no-fit AUROC is not constrained, it can be <0.5, representing performance that’s worse than chance.

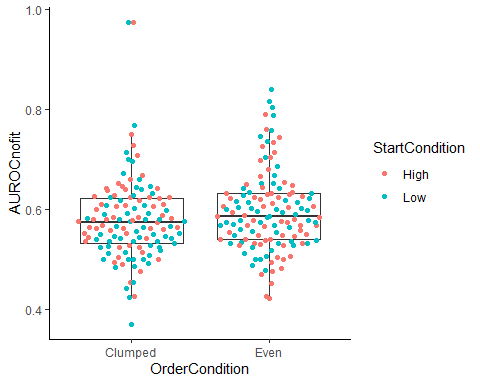
Here we compare this no-fit AUROC to the AUROC obtained from the fit.The black line is the identity. As expected, the fit tends to give a higher AUROC, because the left-hand end of the ROC curve ends at Phit=R, not Phit=0. You can see the 0.5 floor on the y axis. 

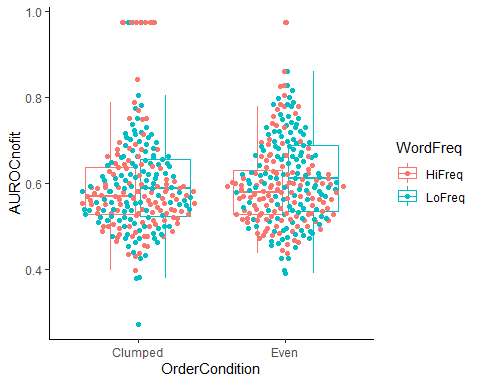
There is 1 ppt with nofit-AUROC<0.4: g3hR3K9F4Bb28KbBMUPn. This person has pHit=0.64 and pFA=0.89, so indeed worse than chance. The two ppts over on the right both have pHit=1 which gives d’=1, even though pFA is also very high. Clearly, the fitted AUROC makes more sense for these ppts.

I thought it would be interesting to check out the agreement between AUROCs estimate from overlapping subsets of the data. This plot shows no-fit AUROC for a given subject, using the high-freq words vs low-freq words. There is a lot of scatter but the correlations are highly significant in both cases. A Wilcoxon rank sum test finds a significant difference in no-fit AUROC between low- and high-frequency words.

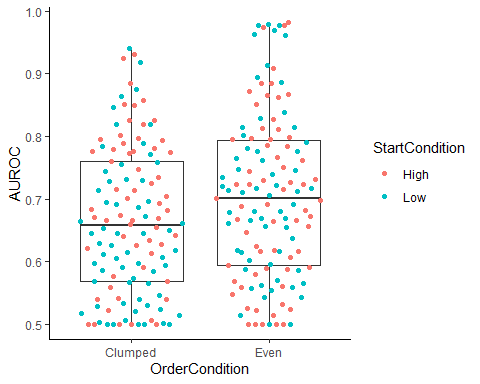
## `geom\_smooth()` using formula 'y ~ x'

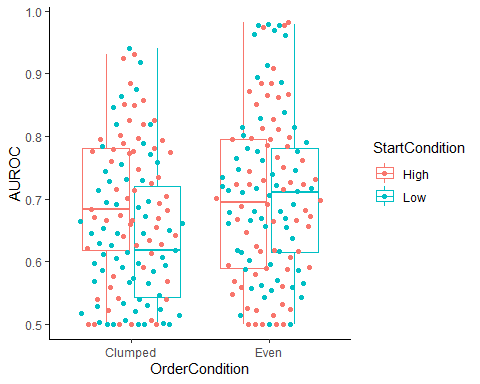


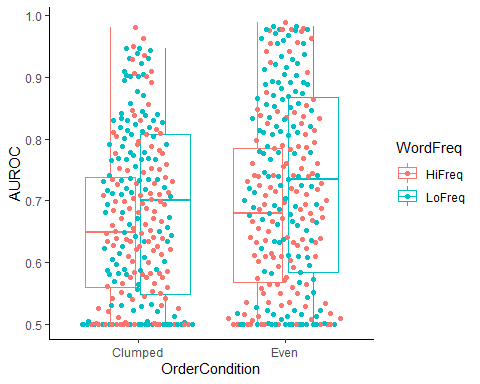
Now let’s look at how this no-fit AUROC varies with word frequency and order condition. This plot shows no-fit AUROC, from the data using all words, plotted by order condition. There is a weak tendency for higher no-fit AUROC in the “even” condition. 

This plot shows no-fit AUROC obtained using only low- or high-freq words, again plotted by order condition. In both cases, dprime is slightly higher for low-frequency words. 

## Overall performance as measured by fitted AUROC

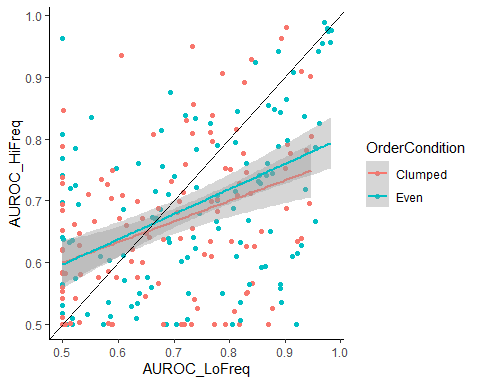
Reassuringly the same general conclusions hold when we look at the fitted AUROC. We first look at overall performance as measured by the area under the receiver operating characteristic curve (fitted using R and d’). Recall that our fitting procedure means that this cannot be below 0.5. This plot shows AUROC, fitted using all words, plotted by order condition. There is a weak tendency for higher AUROC in the “even” condition. 

Here, we plot the same data but now draw separate box plots by start condition, to see if that makes a difference. It does not seem to. 

This plot shows AUROC fitted using only low- or high-freq words, again plotted by order condition. In both cases, dprime is slightly higher for low-frequency words. 

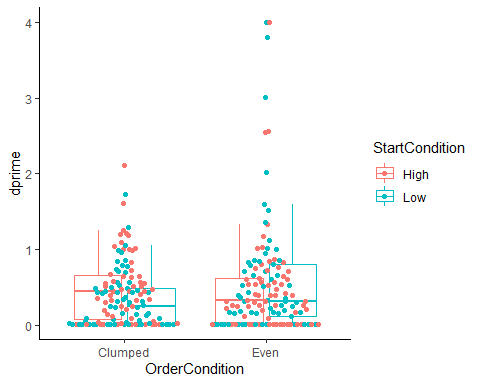
This plot shows AUROC fitted for a given subject, using the high-freq words vs low-freq words. There is a lot of scatter but the correlations are highly significant in both cases. A Wilcoxon rank sum test finds a significant difference in AUROC between low- and high-frequency words.

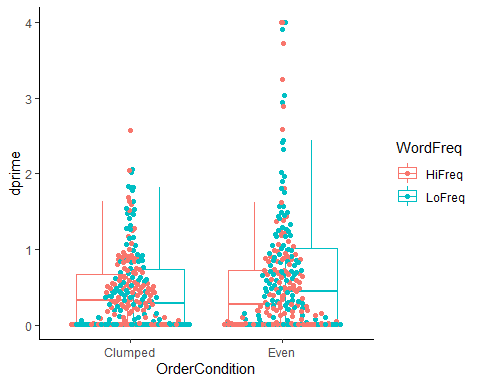
## `geom\_smooth()` using formula 'y ~ x'



## Familiarity parameter Dprime

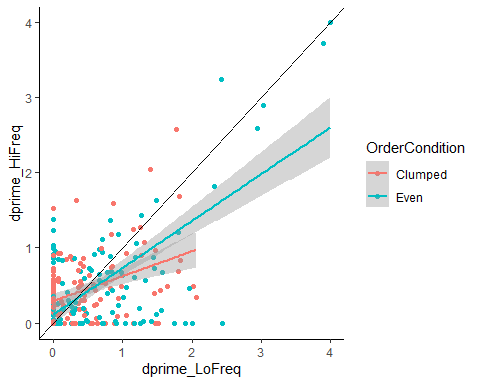
Now we do the same for dprime.

This plot shows dprime, fitted using all words, with separate box plots for order condition. There does not seem to be any effect either of order or of start condition. 

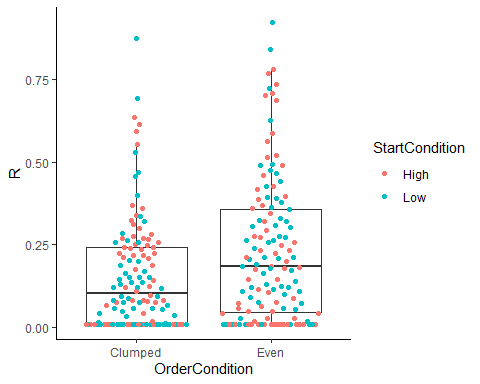
This plot shows dprime fitted using only low- or high-freq words, again plotted by order condition. There does not seem to be any significant effect of word frequency on dprime. 

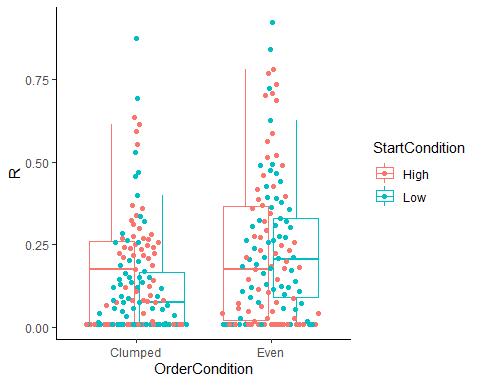
This plot shows dprime fitted for a given subject, using the high-freq words vs low-freq words. There is a lot of scatter but the correlations are highly significant in both cases. A Wilcoxon rank sum test finds no significant difference in dprime between low- and high-frequency words.

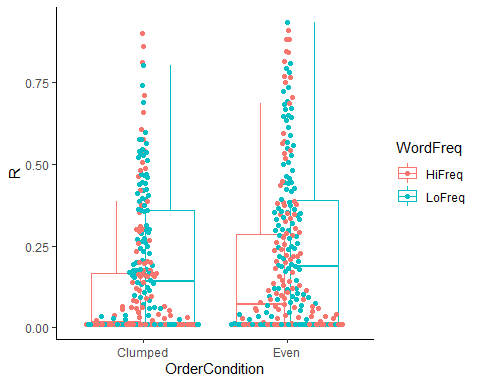
## `geom\_smooth()` using formula 'y ~ x'



## Recollection parameter R

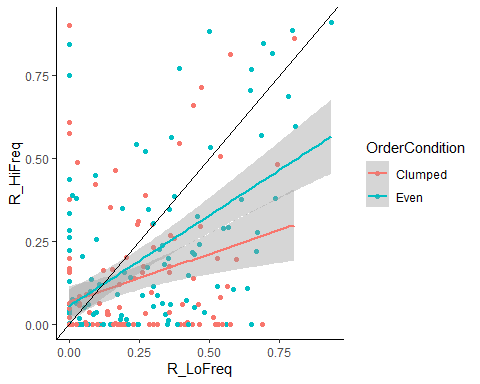
Now we do the same for R. This plot shows R, fitted using all words, plotted by order condition. There is a weak tendency for higher R in the “even” condition. 

Here, we plot the same data but now draw separate box plots for start condition, to see if that makes a difference. It does not seem to. 

This plot shows R fitted using only low- or high-freq words, again plotted by order condition. In both cases, R is convincingly higher for low-frequency words. 

This plot shows R fitted for a given subject, using the high-freq words vs low-freq words. There is a lot of scatter but the correlation is highly significant. A Wilcoxon rank sum test finds a significant difference in R between low- and high-frequency words.

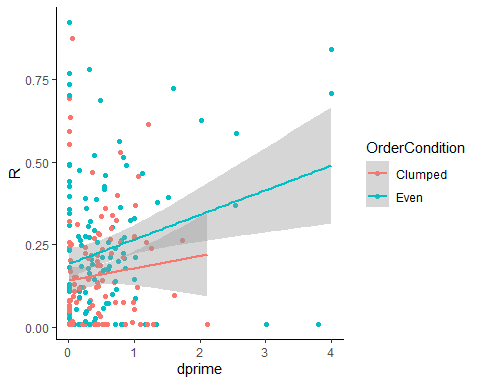
## `geom\_smooth()` using formula 'y ~ x'



## Relationship between R and d’

This shows R vs d’, for fits to all data.

## `geom\_smooth()` using formula 'y ~ x'



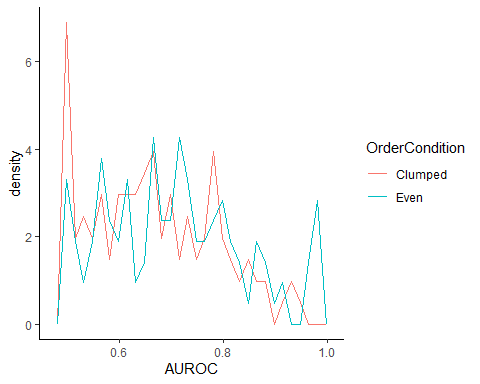
There is a significant positive correlation between the and parameters: Pearson correlation: rho =0.2, p=6.8e-08, Spearman correlation: rho =0.15, p=2.9e-05.

Note that in any individual fit there tends to be a tradeoff between and . So if people were identical and any differences were due to uncertainty in the fit, we would expect a negative correlation. The positive correlation indicates that people are different: those with good recollection also tend to have good . Note that the slope is the same for both groups, but the R is higher for the “even” group, as we have seen.

# Statistical analysis

## AUROC is significantly better for low-frequency words and for even distributions

We explored fitting AUROC with a mixed-effect model, with two between-subjects factors (Order, even vs clumped, and Start, high vs low), and one within-subjects factor (word frequency during exposure phase, high vs low), with participant as a random factor.

This is the overall distribution of the fitted AUROC. 

AUROC is fairly Gaussian except for the peak at its lower bound of 0.5. To obtain something I could reasonably fit with a gamma distribution, I transformed AUROC into AUROCnorm = 2(AUROC-0.5). This is in the range (0-1) whereas our original fitted AUROC was in the range (0.5-1). Again, I replaced values < 0.01 (there were a few due to rounding error etc) with 0.01.

Fitting AUROCnorm with a gamma distribution, I found a highly significant effect of word frequency but no effect of order or start condition:

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: Gamma ( log )  
## Formula: AUROCnorm ~ WordFreq + OrderCondition + (1 | ParticipantIdentifier)  
## Data: results\_ByFreq  
##   
## AIC BIC logLik deviance df.resid   
## 22.0 43.1 -6.0 12.0 493   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.6209 -0.6849 0.1373 0.7002 2.0234   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## ParticipantIdentifier (Intercept) 0.1994 0.4465   
## Residual 0.3803 0.6167   
## Number of obs: 498, groups: ParticipantIdentifier, 249  
##   
## Fixed effects:  
## Estimate Std. Error t value Pr(>|z|)   
## (Intercept) -1.28846 0.08106 -15.895 < 2e-16 \*\*\*  
## WordFreqLoFreq 0.16449 0.06372 2.581 0.00984 \*\*   
## OrderConditionEven 0.16165 0.10256 1.576 0.11498   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) WrdFLF  
## WordFrqLFrq -0.379   
## OrdrCndtnEv -0.643 -0.010

Using a standard linear model, I found a highly significant effect of word frequency and a marginally significant effect of order but again no effect of start condition.

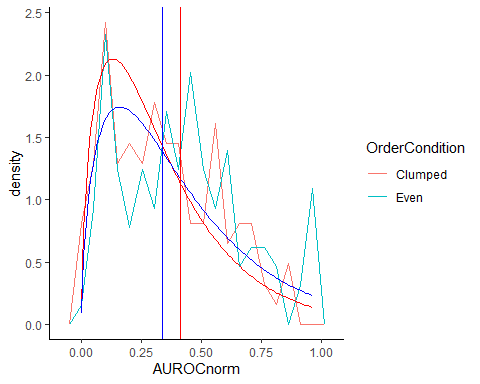
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: AUROC ~ WordFreq + OrderCondition + (1 | ParticipantIdentifier)  
## Data: results\_ByFreq  
##   
## REML criterion at convergence: -565.8  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.22136 -0.64214 -0.02693 0.66872 2.38527   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## ParticipantIdentifier (Intercept) 0.008464 0.0920   
## Residual 0.011515 0.1073   
## Number of obs: 498, groups: ParticipantIdentifier, 249  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 6.588e-01 1.182e-02 3.413e+02 55.741 < 2e-16 \*\*\*  
## WordFreqLoFreq 3.218e-02 9.617e-03 2.480e+02 3.346 0.000947 \*\*\*  
## OrderConditionEven 3.441e-02 1.512e-02 2.470e+02 2.276 0.023707 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) WrdFLF  
## WordFrqLFrq -0.407   
## OrdrCndtnEv -0.652 0.000

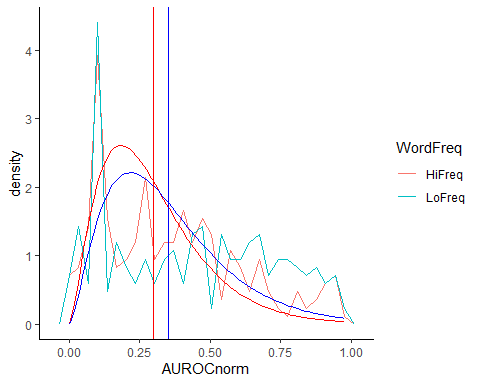
We also examined the effect of order and start condition on the data fitted using all words (thus containing only between-subjects factors). This revealed a significant effect of order, with performance being higher for “even” than for “clumped”. The results were the same whether I fitted with a gamma distribution or a Gaussian.

##   
## Call:  
## glm(formula = AUROCnorm ~ OrderCondition, family = Gamma(link = log),   
## data = results\_AllFreq)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.3118 -0.7438 -0.0528 0.3894 1.1304   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.08054 0.05818 -18.571 <2e-16 \*\*\*  
## OrderConditionEven 0.19984 0.08147 2.453 0.0149 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Gamma family taken to be 0.41303)  
##   
## Null deviance: 170.31 on 248 degrees of freedom  
## Residual deviance: 167.83 on 247 degrees of freedom  
## AIC: -13.188  
##   
## Number of Fisher Scoring iterations: 5

Overall, then, I think it’s fair to conclude that AUROC is significantly better for low-frequency words and for even distributions.

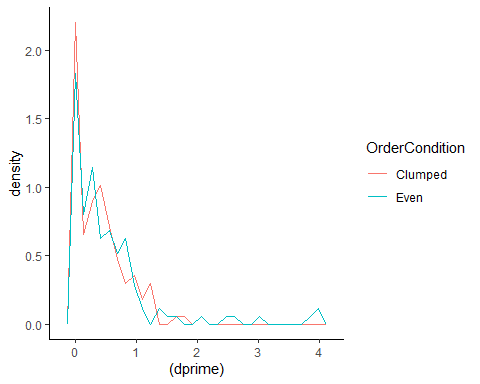
### Gamma-distribution fits for AUROC

I thought it would be worth looking at what the model fit implies. Here it is for the fits to all data, where we found a significant effect of order. The frequency polygons show the distribution of fitted AUROC-norm for the two cases. The smooth curves show the fitted gamma distributions. The vertical lines mark the means of these. 

Here is a similar plot for the fits separately to low and high frequency words, where we found a significant effect of frequency: 

I feel that these gamma fits are clearly better than a Gaussian would be, and this justifies the choice of this family.

## d’ shows no main effects but there’s an interaction between order and frequency

This plot shows the distribution of d’. 

It is highly skewed and has a peak at its floor. It therefore seems sensible to fit it with a gamma distribution.

I had difficulty getting this to converge, but it did so if I changed to the “optimx” optimizer.

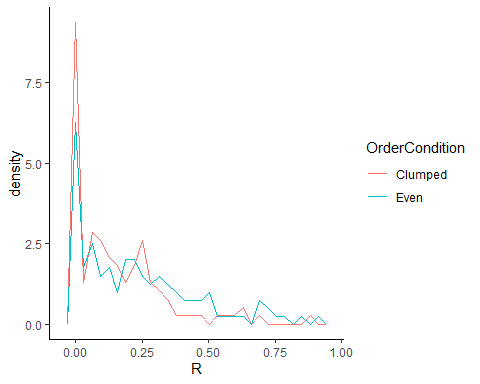
I find no main effect of word frequency, start condition or order condition , but I do find a significant interaction between word frequency and order condition:

## Loading required namespace: optimx

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: Gamma ( log )  
## Formula: dprime ~ WordFreq \* OrderCondition + (1 | ParticipantIdentifier)  
## Data: results\_ByFreq  
## Control:   
## glmerControl(optimizer = "optimx", calc.derivs = FALSE, optCtrl = list(method = "bobyqa"))  
##   
## AIC BIC logLik deviance df.resid   
## 269.0 294.3 -128.5 257.0 492   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.05836 -0.68661 -0.02512 0.61342 1.66513   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## ParticipantIdentifier (Intercept) 1.5977 1.2640   
## Residual 0.8689 0.9321   
## Number of obs: 498, groups: ParticipantIdentifier, 249  
##   
## Fixed effects:  
## Estimate Std. Error t value Pr(>|z|)   
## (Intercept) -1.353693 0.142188 -9.520 < 2e-16 \*\*\*  
## WordFreqLoFreq -0.173114 0.119347 -1.451 0.14692   
## OrderConditionEven 0.005614 0.199095 0.028 0.97750   
## WordFreqLoFreq:OrderConditionEven 0.434345 0.167113 2.599 0.00935 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) WrdFLF OrdrCE  
## WordFrqLFrq -0.420   
## OrdrCndtnEv -0.714 0.300   
## WrdFrLF:OCE 0.300 -0.714 -0.420

dprime is significantly higher for low frequency words in the even condition than for the other three situations (low-freq and clumped, or high-freq).

## R is significantly better for low-frequency words and for even distributions

Like d’, R is highly skewed so I again decided to fit it with a gamma distribution: 

This does converge for fitting R. In the fits done separately by frequency, we find a highly significant effect of word frequency, but a marginally non-significant effect of order:

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: Gamma ( log )  
## Formula: R ~ WordFreq + OrderCondition + (1 | ParticipantIdentifier)  
## Data: results\_ByFreq  
##   
## AIC BIC logLik deviance df.resid   
## -832.7 -811.7 421.4 -842.7 493   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.1106 -0.6477 -0.1828 0.5990 2.1452   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## ParticipantIdentifier (Intercept) 1.4089 1.1870   
## Residual 0.7811 0.8838   
## Number of obs: 498, groups: ParticipantIdentifier, 249  
##   
## Fixed effects:  
## Estimate Std. Error t value Pr(>|z|)   
## (Intercept) -2.81795 0.13785 -20.442 < 2e-16 \*\*\*  
## WordFreqLoFreq 0.55486 0.08589 6.460 1.04e-10 \*\*\*  
## OrderConditionEven 0.32243 0.18263 1.766 0.0775 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) WrdFLF  
## WordFrqLFrq -0.319   
## OrdrCndtnEv -0.680 0.014

I also explored fitting log(R) with a linear model. Fitting log(R) shows a highly significant effect of word frequency, and a marginally significant effect of order:

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: log(R) ~ WordFreq + OrderCondition + (1 | ParticipantIdentifier)  
## Data: results\_ByFreq  
##   
## REML criterion at convergence: 1860.9  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -1.8762 -0.8337 0.1216 0.6700 2.1443   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## ParticipantIdentifier (Intercept) 0.9389 0.969   
## Residual 1.6714 1.293   
## Number of obs: 498, groups: ParticipantIdentifier, 249  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) -3.1596 0.1338 355.2910 -23.615 < 2e-16 \*\*\*  
## WordFreqLoFreq 0.5400 0.1159 248.0000 4.660 5.16e-06 \*\*\*  
## OrderConditionEven 0.3701 0.1689 247.0000 2.192 0.0293 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) WrdFLF  
## WordFrqLFrq -0.433   
## OrdrCndtnEv -0.644 0.000

When we look at the fits to all data, the effect of order is significant whether we fit R with a gamma distribution, or log(R) with a Gaussian.

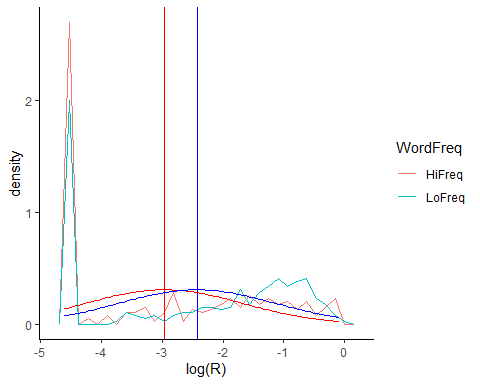
##   
## Call:  
## glm(formula = R ~ OrderCondition, family = Gamma(link = log),   
## data = results\_AllFreq)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.0929 -1.7231 -0.2498 0.4667 2.3940   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.85353 0.09099 -20.370 < 2e-16 \*\*\*  
## OrderConditionEven 0.39558 0.12741 3.105 0.00213 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Gamma family taken to be 1.010104)  
##   
## Null deviance: 377.10 on 248 degrees of freedom  
## Residual deviance: 367.45 on 247 degrees of freedom  
## AIC: -322.39  
##   
## Number of Fisher Scoring iterations: 6

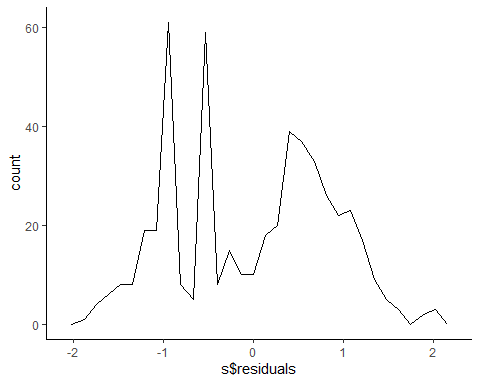
##   
## Call:  
## lm(formula = log(R) ~ OrderCondition, data = results\_AllFreq)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.4389 -1.6246 0.4627 1.1772 2.4888   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.6221 0.1308 -20.047 <2e-16 \*\*\*  
## OrderConditionEven 0.4558 0.1831 2.489 0.0135 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.445 on 247 degrees of freedom  
## Multiple R-squared: 0.02446, Adjusted R-squared: 0.02051   
## F-statistic: 6.194 on 1 and 247 DF, p-value: 0.01348

So, I feel fairly safe in concluding that there are significant effects both of word frequency and order R is higher for low-frequency words and for the even distribution. I think the order effect probably didn’t show up in the mixed-model just because there was more noise when R is fitted to less data.

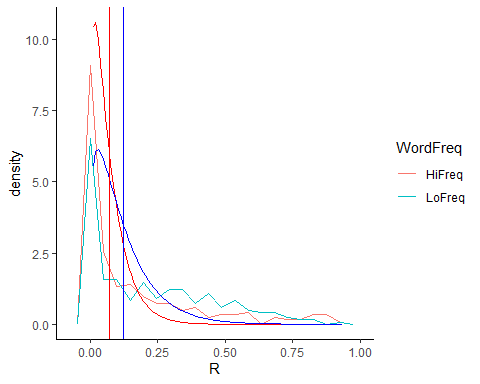
## Comparing Gaussian and Gamma-distribution fits for R

I wanted to justify my choice of a gamma distribution.

I feel we have to be a bit careful about fitting a Gaussian model. Here is the distribution of log(R) for a linear mixed-model fitted to the by-frequency data, log(R) ~ WordFreq + (1|ID), with the fitted Gaussians marked on (vertical lines mark the means): 

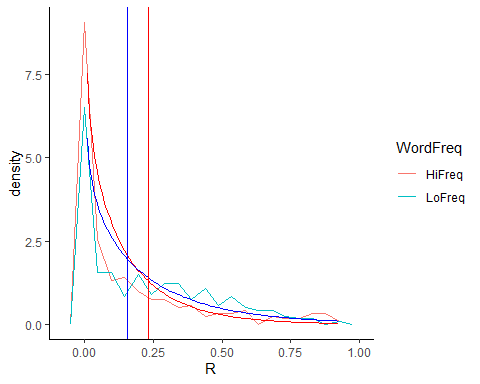
The model is thus a poor description of the data. It’s had to put the mean of the Gaussians in between the two peaks, so the residuals are thus quite non-normal, contradictory to the model assumptions: 

The distribution of residuals has a central dip and then peaks on either side, instead of a central peak.

Here are the fitted distributions for one of the gamma models. This is “R ~ OrderCondition” for the The frequency polygons show the distribution of fitted R for the two cases. The smooth curves show the fitted gamma distributions. The vertical lines mark the means of these. 

This model may not be perfect, but I think it’s clear it’s a far better description of the data, as reflected in its far lower AIC (for the same number of fit parameters)

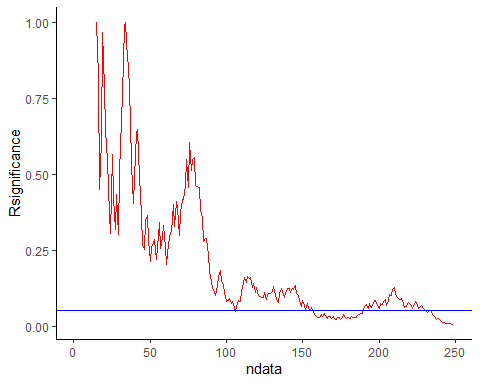
Here is the same thing for “R ~ OrderCondition” fitted to all data.



It’s obvious that this is a far better fit than a Gaussian would be!

## Check robustness

Tracking how significance varies as a function of the amount of data collected. Here, I’m plotting the significance reported by a Wilcoxon test comparing R in the even vs clumped condition, for the first N subjects (N plotted along x-axis). I don’t know if the number of subjects was set in advance but want to check the result isn’t very sensitive to exactly when we stopped.



A little bit concerning that had data collection stopped slightly earlier, the result would be NS?

## Conclusions

Overall performance is higher for even word order and for low-frequency words. Fitting the ROC curve suggests that this improvement is due to improved recollection, not familiarity.