Analysing gender balance in video game dialogue

Introduction

This section looks at the proportion of dialogue in video games for female and male characters. The tests below test:

- Descriptive statistics of gender balance
- Change over time
- Statistical difference from hypothetical baselines.

The amount of dialogue can be measured in several different ways, including the number of syllables, words, sentences and script lines. We prefer a measure in words because this measure is:

- Reliable: Words are not totally straightforward to measure, but there are standard solutions. In contrast, estimates of syllables and sentences rely on additional assumptions.
- Valid: Words are a more valid measure of text length than the number of lines, because a line can vary from one word to several sentences.

Basic statistics

Load libraries

```
library(ggplot2)
library(ggrepel)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(grid)
library(rjson)
library(tidyr)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-40. For overview type 'help("mgcv-package")'.
```

```
library(tidymv)
```

```
## tidymv will be deprecated. Users are recommended
##     to check out the in-progress replacement tidygam
##     (https://github.com/stefanocoretta/tidygam).
```

```
library(ggpubr)
```

Load statistics for all groups for all games and work out proportion of female dialogue compared to all female and male dialogue for different measures of length.

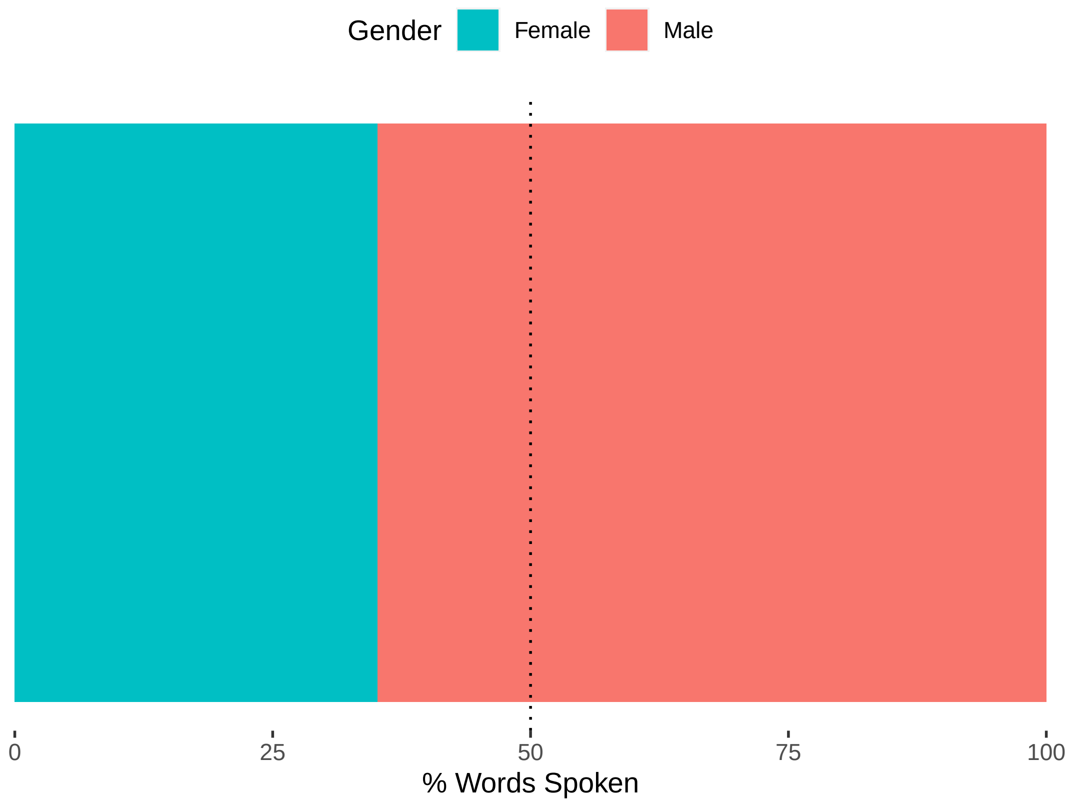
```
stats = read.csv("../results/generalStats.csv",stringsAsFactors = F)
# Remove alternative measures
stats = stats[stats$alternativeMeasure!="True",]
stats = stats[!is.na(stats$words),]
d = NULL
folders = unique(stats$folder)
for(folder in folders){
  sxM = stats[stats$folder==folder & stats$group == "male",]
  sxF = stats[stats$folder==folder & stats$group == "female",]
  js = fromJSON(file = paste0(folder,"meta.json"))
  if(nrow(sxM)>0 & nrow(sxF)>0){
    d = rbind(d,
      data.frame(
        folder = folder,
        series = sxF$series,
        game = sxF$game,
        lines = sxF$lines / (sxF$lines + sxM$lines),
        words = sxF$words / (sxF$words + sxM$words),
        sentences = sxF$sentences / (sxF$sentences + sxM$sentences),
        syllables = sxF$syllables / (sxF$syllables + sxM$syllables),
        fw = sxF$words,
        mw = sxM$words,
        year = js$year,
        shortName = tail(strsplit(folder,"/")[[1]],1),
        femCharProp = sxF$numCharacters / (sxF$numCharacters+sxM$numCharacters),
        maleWordsPerChar = 1000* ((sxM$words/(sxM$words+sxF$words)) / sxM$numCharacters),
        femaleWordsPerChar = 1000* ((sxF$words/(sxM$words+sxF$words)) / sxF$numCharacters)
      ))
  }
}
shortNameChanges = list(
  c("KingdomHearts", "KH"),
  c("KingsQuest", "KQ"),
  c("_Remake", "-R"),
  c("TheSecretOfMonkeyIsland", "MI1"),
  c("MonkeyIsland2", "MI2"),
  c("TheCurseOfMonkeyIsland", "MI3"),
  c("SuperMarioRPG", "SMario"),
  c("FFX_B", "FFX"),
  c("MassEffect1", "ME1"),
  c("MassEffect2", "ME2"),
  c("MassEffect3C", "ME3"),
  c("B$", ""),
  c("DS$", ""),
  c("StarWarsKOTOR","KOTOR"),
  c("DragonAgeOrigins","DAO"),
  c("_","")
)
for(snc in shortNameChanges){
  d$shortName = gsub(snc[1],snc[2],d$shortName)
}
```

Overall proportion of female dialogue:

```
allGamesPropFemale = (sum(stats[stats$group=="female",]$words,na.rm=T)/
                        (sum(stats[stats$group=="female",]$words,na.rm=T)+
                           sum(stats[stats$group=="male",]$words,na.rm=T)))*100
allGamesPropFemale
```

```
## [1] 35.16371
```

```
allGamesPropMale = 100-allGamesPropFemale
dx = data.frame(
  Gender=factor(c("Male","Female"),levels=c("Male","Female")),
  percentageWords=c(allGamesPropMale,allGamesPropFemale))
mainStatGraph = ggplot(dx,aes(x=1,y=percentageWords,fill=Gender))+ geom_bar(stat='identity')+
  geom_hline(yintercept=50,linetype="dotted") +
  coord_flip(ylim = c(0,100)) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        legend.position = "top") +
  scale_fill_discrete(breaks=c("Female","Male"),name="Gender")+
  ylab("% Words Spoken") +
  xlab("")
mainStatGraph
```



```
pdf('../results/graphs/OverallWordsByGender.pdf',width=6,height=3)
mainStatGraph
dev.off()
```

```
## pdf
##   2
```

Plots for specific series:

```
seriesToPlot = names(table(d$series)[table(d$series)>1])
for(series in seriesToPlot){
  dx = d[d$series==series,]
  dx$game = factor(dx$game, levels = unique(dx$game[order(dx$year,decreasing = T)]))
  dx$words.m = 1 - dx$words
  sx = pivot_longer(dx,c("words","words.m"))
  sx$group = factor(sx$name,
                    levels=c("words.m","words"),
                    labels=c("Male","Female"))
  sx$measurement = sx$value*100

  gx = ggplot(sx, aes(x=game,y=measurement,fill=group)) +
    geom_bar(stat='identity')+
    geom_hline(yintercept=50,linetype="dotted") +
    coord_flip(ylim = c(0,100)) +
    theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.background = element_blank(),
          legend.position = "top") +
    scale_fill_discrete(breaks=c("Female","Male"),name="Gender")+
    ylab("% Words Spoken") +
    xlab("")
  gx

  # write
  fileName = paste0("../results/graphs/series/",series,".pdf")
  fileName = gsub(" ","_",fileName)
  pdf(fileName,width=5,height=4)
  gx
  dev.off()
}
```

Summary of proportion of female dialogue according to different measures:

```
knitr::kable(d[,c("game","syllables","words","sentences","lines")],digits = 3)
```

game	syllables	words	sentences	lines
The Elder Scrolls III: Morrowind	0.328	0.332	0.343	0.353
The Elder Scrolls IV: Oblivion	0.408	0.408	0.413	0.422
The Elder Scrolls II: Daggerfall	0.215	0.212	0.200	0.174
The Elder Scrolls V: Skyrim	0.306	0.305	0.305	0.305
Final Fantasy V	0.267	0.270	0.312	0.329
Final Fantasy X	0.323	0.322	0.318	0.303
Final Fantasy X-2	0.478	0.484	0.548	0.606
Final Fantasy XV	0.200	0.195	0.144	0.113
Final Fantasy XIV	0.333	0.334	0.341	0.358
Final Fantasy VIII	0.318	0.322	0.299	0.320
Final Fantasy VII Remake	0.369	0.372	0.383	0.368
Final Fantasy	0.230	0.229	0.242	0.259
Final Fantasy VI	0.180	0.182	0.234	0.219
Final Fantasy IV	0.106	0.106	0.127	0.136
Lightning Returns: Final Fantasy XIII	0.542	0.546	0.581	0.606
Final Fantasy XII	0.273	0.272	0.271	0.262
Final Fantasy IX	0.262	0.262	0.273	0.270
Final Fantasy XIII-2	0.413	0.412	0.411	0.408
Final Fantasy XIII	0.419	0.421	0.444	0.454
Final Fantasy II	0.297	0.298	0.296	0.274
Final Fantasy VII	0.258	0.259	0.272	0.282

game	syllables	words	sentences	lines
Super Mario RPG: Legend of the Seven Stars	0.166	0.166	0.181	0.161
Star Wars: Knights of the Old Republic	0.280	0.286	0.293	0.283
Chrono Trigger	0.375	0.380	0.397	0.416
Horizon Zero Dawn	0.441	0.444	0.482	0.535
Monkey Island 2: LeChuck's Revenge	0.172	0.168	0.159	0.131
The Secret of Monkey Island	0.079	0.079	0.089	0.067
The Curse of Monkey Island	0.069	0.067	0.060	0.052
King's Quest I: Quest for the Crown	0.123	0.112	0.088	0.118
King's Quest VI	0.062	0.064	0.071	0.059
King's Quest VIII	0.142	0.142	0.143	0.113
King's Quest VII: The Princeless Bride	0.490	0.492	0.529	0.562
King's Quest Chapters	0.344	0.349	0.355	0.363
King's Quest V	0.295	0.293	0.277	0.249
King's Quest II: Romancing the Throne	0.797	0.798	0.689	0.692
King's Quest III: To Heir Is Human	0.239	0.239	0.223	0.233
King's Quest IV: The Perils of Rosella	0.807	0.800	0.779	0.754
Kingdom Hearts	0.228	0.221	0.199	0.182
Kingdom Hearts II	0.155	0.155	0.150	0.143
Kingdom Hearts III	0.139	0.140	0.135	0.126
Kingdom Hearts 3D: Dream Drop Distance	0.067	0.067	0.064	0.070
Dragon Age 2	0.400	0.401	0.415	0.409
Dragon Age: Origins	0.322	0.322	0.318	0.314
Persona 3	0.431	0.430	0.459	0.481
Persona 4	0.478	0.476	0.498	0.487
Persona 5	0.396	0.396	0.421	0.401
Mass Effect 3	0.407	0.403	0.407	0.401
Mass Effect	0.417	0.418	0.415	0.412
Mass Effect 2	0.441	0.435	0.405	0.415
Stardew Valley	0.450	0.452	0.427	0.461

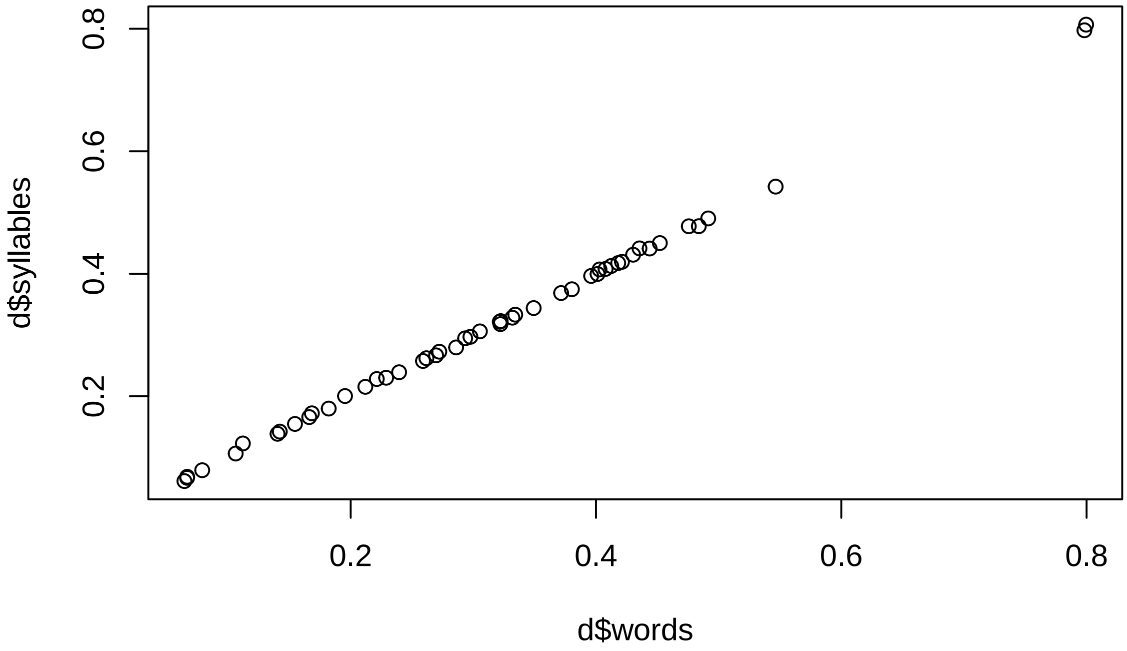
Write some stats for the paper:

```
# Write general stats
minFemaleProp = d[d$words==min(d$words),]
maxFemaleProp = d[d$words==max(d$words),]

minFemalePropStr = paste0(round(100*minFemaleProp$words),"\\% (\\emph{",minFemaleProp$game,"})")
maxFemalePropStr = paste0(round(100*maxFemaleProp$words),"\\% (\\emph{",maxFemaleProp$game,"})")

cat(minFemalePropStr,file="../results/latexStats/gameWithMinimumFemaleWords.tex")
cat(maxFemalePropStr,file="../results/latexStats/gameWithMaximumFemaleWords.tex")

cat(round(100*d[d$game=="Final Fantasy XV",]$words),
 file = "../results/latexStats/FFXV_PropFemaleDialogue.tex")
```

The proportion of female dialogue ranges from 6% (*King's Quest VI*) to 80% (*King's Quest IV: The Perils of Rosella*).

The estimates from different measures of length are highly correlated:

```
plot(d$words,d$syllables)
```



```
cor(d$words,d$syllables,method = 'k')
```

```
## [1] 0.9967347
```

```
cor(d$words,d$sentences)
```

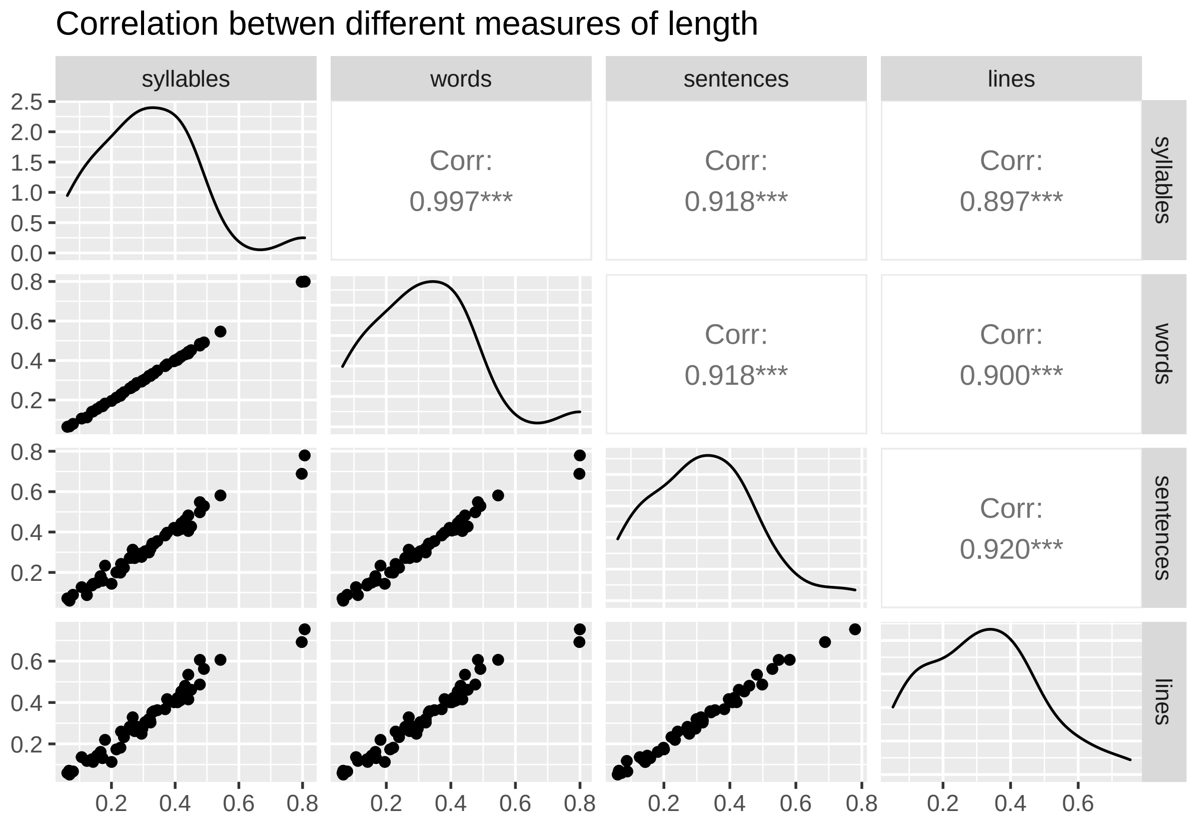
```
## [1] 0.9857857
```

```
cor(d$syllables,d$sentences)
```

```
## [1] 0.983933
```

Correlation between all measures, using Kendall's correlation for significance:

```
measures = c("syllables","words","sentences","lines")
corK = function(data,mapping,...){
  ggally_cor(data,mapping,method="kendall")
  }
ggpairs(d[,measures],
        upper = list(continuous = corK),
        title="Correlation betwen different measures of length")
```



What are the maximum differences between measures of syllables, words and sentences?

```
mx = c("syllables","words","sentences")
d$maxDiff = apply(d[,mx],1,function(X){max(abs(outer(X,X,"-")))})
```

For 95% of games, the measures of syllables, words, and sentences were within 5.5 percentage points of each other. Two games had higher differences. The first is King's Quest 2, which has a very small amount of dialogue, so small differences in counts can lead to large differences in proportions.

The second is Final Fantasy X-2 (7 percentage points difference), which has a higher estimate for sentences than for words or syllables.

```
x2 = read.csv("../data/FinalFantasy/FFX2/stats_by_character.csv",stringsAsFactors = F)
x2 = x2[x2$group %in% c('male','female'),]
summary(lm(words~lines*group, data=x2[x2$group %in% c("male","female"),]))
```

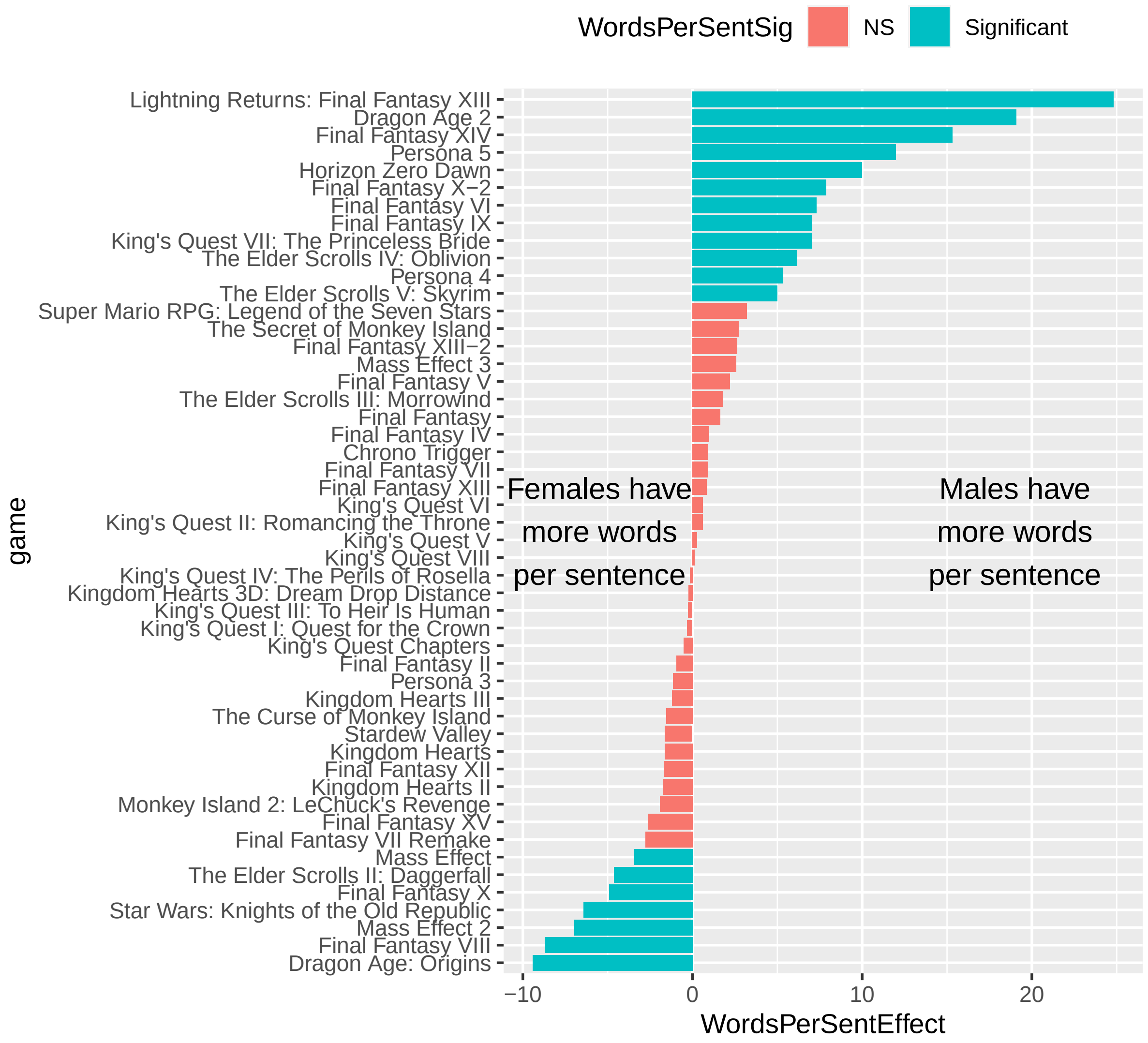
```
##
## Call:
## lm(formula = words ~ lines * group, data = x2[x2$group %in% c("male",
##     "female"), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -698.25  -60.89  -15.50   -1.02 1278.82
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       84.2100    39.9749   2.107   0.0374 *
## lines              6.7689     0.2051  32.996  < 2e-16 ***
## groupmale        -80.0665    46.6103  -1.718   0.0886 .
## lines:groupmale    5.9070     0.9604   6.151 1.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 188.4 on 111 degrees of freedom
## Multiple R-squared:  0.9248, Adjusted R-squared:  0.9227
## F-statistic: 454.7 on 3 and 111 DF,  p-value: < 2.2e-16
```

While the overall proportion of female dialogue is high, FFX-2 has large gender differences in the number of words per sentence. Female speak on average 3.4 words per sentence, and males speak 4.4. We note that the Dale-Chall Readability is also higher for males than females, which agrees with this.

We note that, across games, there is considerable variation in the gender differences in words per sentence:

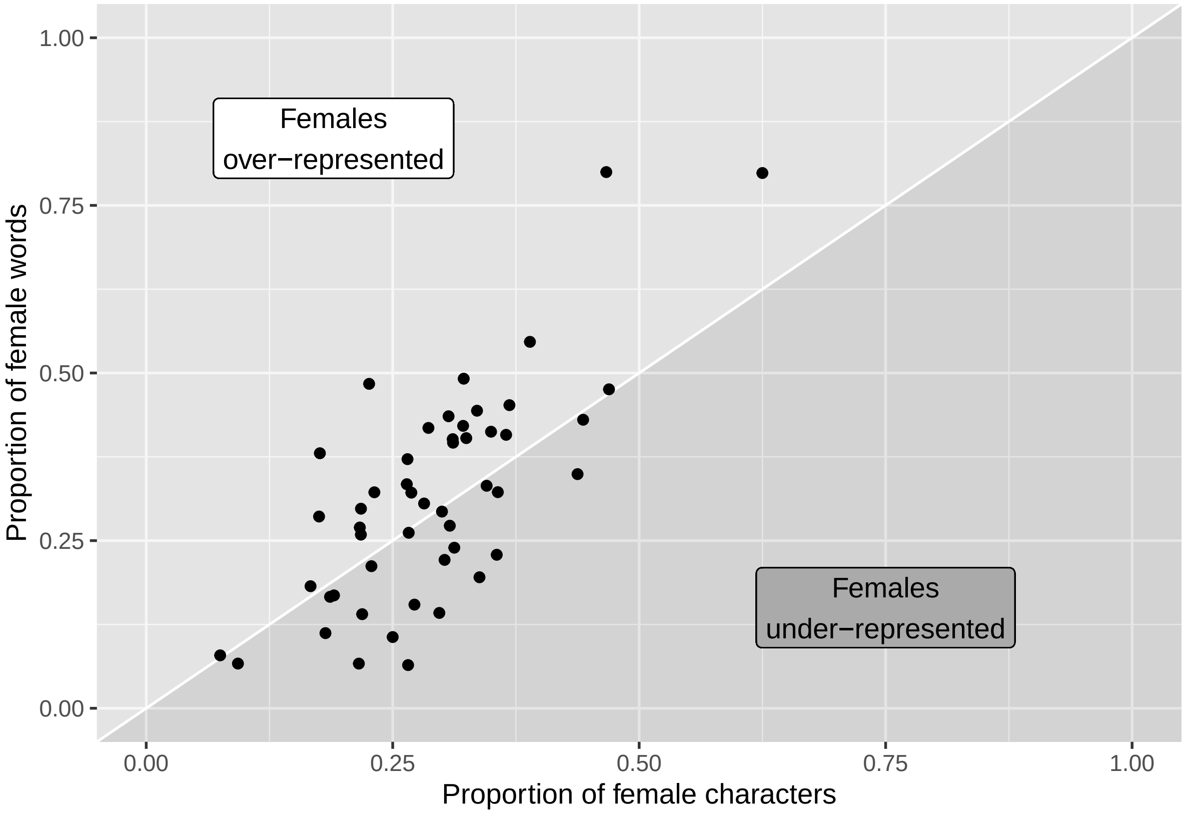
```r
d$WordsPerSentEffect = NA
d$WordsPerSentSig = NA
for(folder in folders){
  x = read.csv(paste0(folder,"stats_by_character.csv"),stringsAsFactors = F)
  if(nrow(x)>0){
    x = x[x$group %in% c("male","female"),]
    sx = summary(lm(words~sentences*group, data=x))
    t.val = sx$coefficients[4,3]
    sig = sx$coefficients[4,4]< (0.05/nrow(d))
    sig = c("NS","Significant")[sig+1]
    d[match(folder,d$folder),]$WordsPerSentEffect = t.val
    d[match(folder,d$folder),]$WordsPerSentSig = sig
  }
}
d$game = factor(d$game,levels=d$game[order(d$WordsPerSentEffect)])

a1 = grobTree(textGrob("Males have\nmore words\nper sentence", x=0.8,  y=0.5))
a2 = grobTree(textGrob("Females have\nmore words\nper sentence", x=0.15,  y=0.5))
ggplot(d,aes(x=WordsPerSentEffect,y=game,fill=WordsPerSentSig))+
  geom_bar(stat="identity") +
  annotation_custom(a1)+
  annotation_custom(a2)+
  theme(legend.position = "top")
```



Proportion of characters

Relationship between dialogue proportions and character proportions:

```
poly = data.frame(
  x = c(-1,2,2),
  y = c(-1,-1,2),
  id = c(1,1,1))

bgColours = c("#458fc2","#ec0085")
bgColours = c("#000000","#AAAAAA")

wordsVCharacters = ggplot(d,aes(y=words,x=femCharProp)) +
  geom_polygon(data = data.frame(
    x = c(-1,2,2,-1,-1,2),
    y = c(-1,-1,2,-1,2,2),
    grp = c("a","a","a","b",'b','b')),
              aes(x = x, y = y,fill=grp),alpha=0.1)+
  scale_fill_manual(breaks=c('a','b'),values=bgColours)+
  coord_cartesian(ylim=c(0,1),xlim=c(0,1))+
  xlab("Proportion of female characters") +
  ylab("Proportion of female words") +
  geom_abline(intercept=0,slope=1,colour="white") +
  geom_label(label="Females\nover-represented",x=0.19,y=0.85) +
  geom_label(label="Females\nunder-represented",x=0.75,y=0.15, fill="#AAAAAA")+
  theme(legend.position = 'none')

wordsVCharacters + geom_point()
```



```
pdf("../results/graphs/WordsVsCharacters_noPoints.pdf",width=4,height = 3.5)
  wordsVCharacters + geom_point(alpha=0)
dev.off()
```

```
## pdf
##   2
```

```
pdf("../results/graphs/WordsVsCharacters.pdf", width=4,height = 3.5)
  wordsVCharacters + geom_point()
dev.off()
```

```
## pdf
##   2
```

There is a significant correlation between the proportion of female characters and the proportion of female dialogue:

```
cor.test(d$words,d$femCharProp)
```

```
##
##  Pearson's product-moment correlation
##
## data:  d$words and d$femCharProp
## t = 7.5226, df = 48, p-value = 1.168e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5749401 0.8416053
## sample estimates:
```

```
##       cor
## 0.7355718
```

Words per character

Is there a difference in the average amount of dialogue given to the average male character compared to the average female character? One confounding aspect of the data is that different games have different amounts of dialogue. So instead of raw words per character, for each game we calculate the average number of words given to a character *per 1000 words of dialogue*. That is, for every 1000 words of dialogue observed, how many go to a single male character on average? And how many to a single female character on average?

On average, a female character says 17.2 words per 1000 words of dialogue.

On average, a male character says 15.3 words per 1000 words of dialogue.

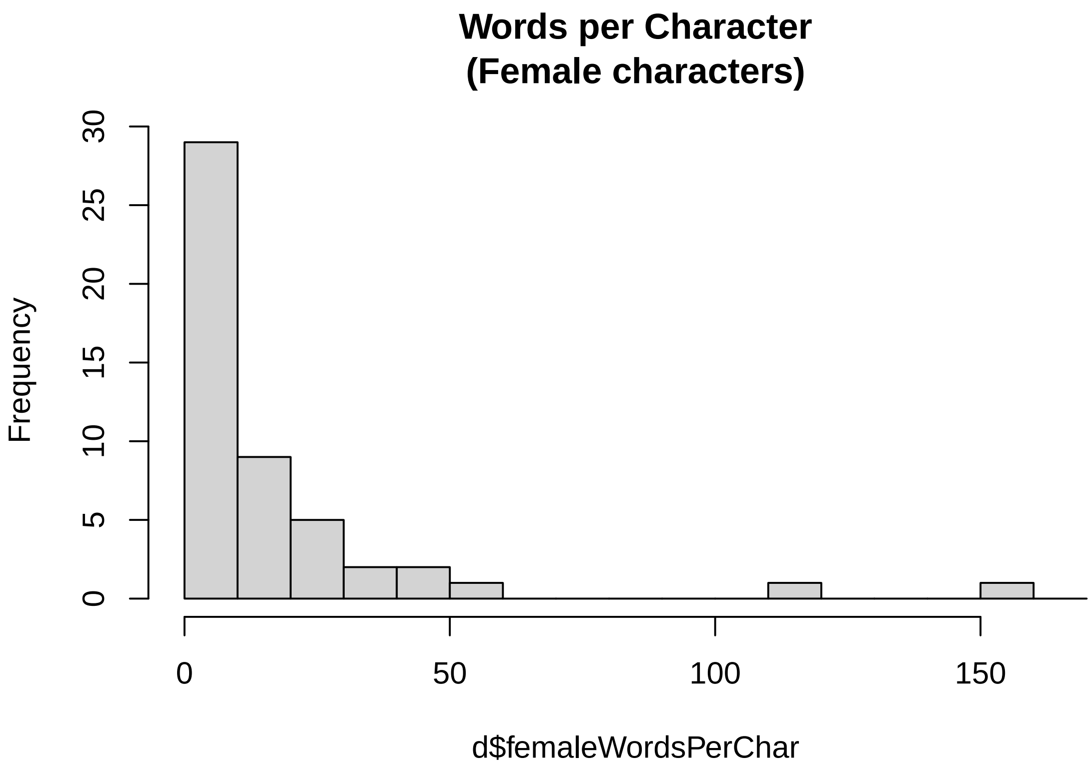
That is, the average female character says slightly more than the average male character. This difference is not significant:

```
t.test(d$femaleWordsPerChar,d$maleWordsPerChar,paired = T)
```

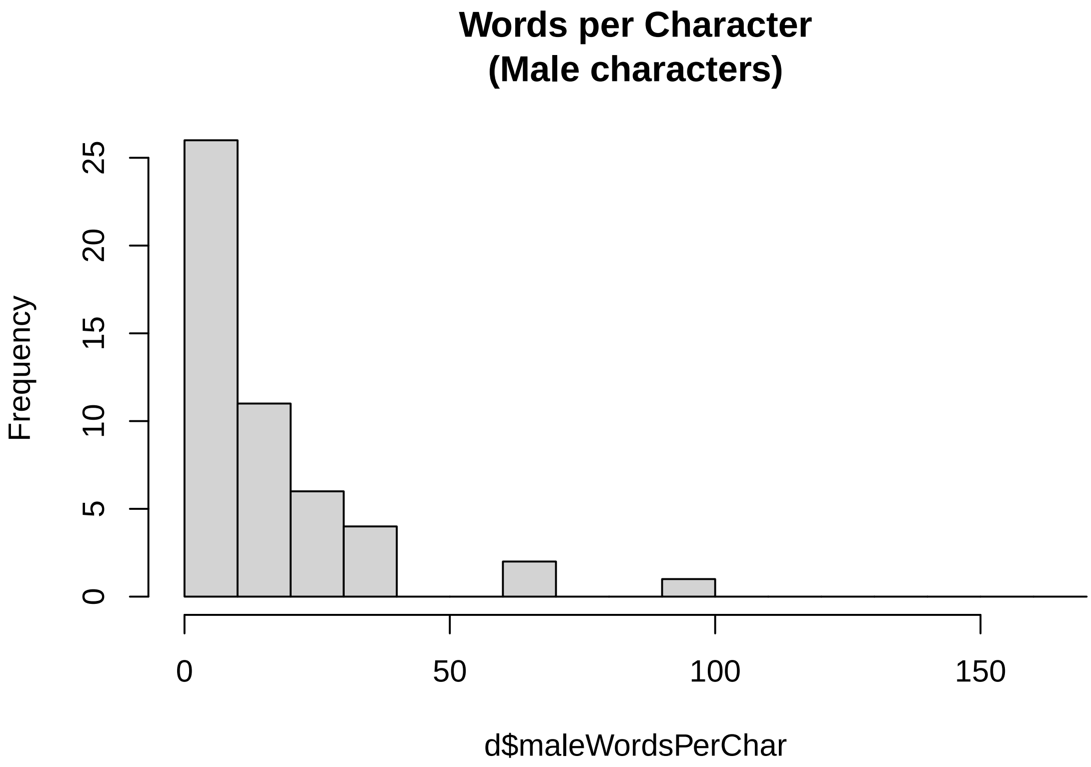
```
##
##  Paired t-test
##
## data:  d$femaleWordsPerChar and d$maleWordsPerChar
## t = 0.64566, df = 49, p-value = 0.5215
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.984454  7.756792
## sample estimates:
## mean of the differences
##                1.886169
```

Here is the distribution over games:

```
hist(d$femaleWordsPerChar, main="Words per Character\n(Female characters)",
     breaks = seq(0,175,by=10))
```



```
hist(d$maleWordsPerChar, main="Words per Character\n(Male characters)",
     breaks = seq(0,175,by=10))
```

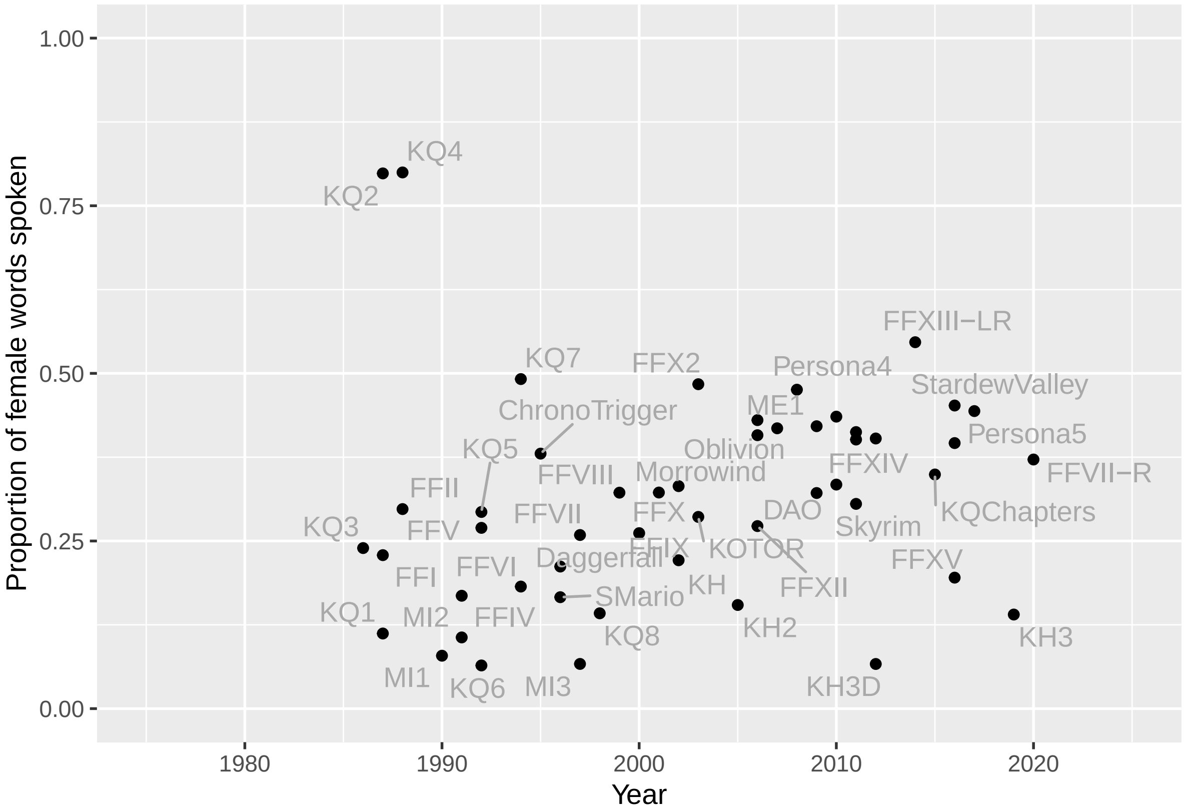


Change over time

This section tests whether the proportion of words spoken by female characters has changed over time. First we can visualise the data:

```
ggplot(d,
       aes(x=year,y=words)) +
  geom_point() +
  coord_cartesian(ylim=c(0,1),xlim=c(1975,2025))+
  ylab("Proportion of female words spoken") +
  xlab("Year") +
  geom_text_repel(aes(label=shortName),color="dark gray")
```

```
## Warning: ggrepel: 7 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



There are clearly two outliers: King's Quest 1 and King's Quest 2. These both have very little dialogue.

Test relationship between proportion of female characters and time:

```
cor.test(d$words,d$year)
```

```
##
##  Pearson's product-moment correlation
##
## data:  d$words and d$year
## t = 1.1528, df = 48, p-value = 0.2547
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1196790  0.4231517
## sample estimates:
##       cor
## 0.1641365
```

There is no significant relationship overall. However, removing the two outliers shows a significant positive correlation:

```
dNoOutliers = d[!d$shortName %in% c("KQ2","KQ4"),]
propFemaleOverTime = cor.test(dNoOutliers$words,dNoOutliers$year)
propFemaleOverTime
```

```
##
##  Pearson's product-moment correlation
##
## data:  dNoOutliers$words and dNoOutliers$year
## t = 3.6394, df = 46, p-value = 0.0006897
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2179794 0.6673123
## sample estimates:
##       cor
## 0.4728292
```

Let's contextualise the linear relationship:

```
# Increase in prop female over time
pm = lm(words~year, data=dNoOutliers)
incPropFemalePerDecade = round((pm$coefficients['year']*100) * 10,1)
incPropFemalePerDecade
```

```
## year
##  6.3
```

```
# Year of parity
parity = ceiling((0.5 - pm$coefficients[1])/pm$coefficients[2])
```

The proportion of female dialogue is increasing by 6.3 percentage points per decade. If this rate continues, parity will be reached by 2036.

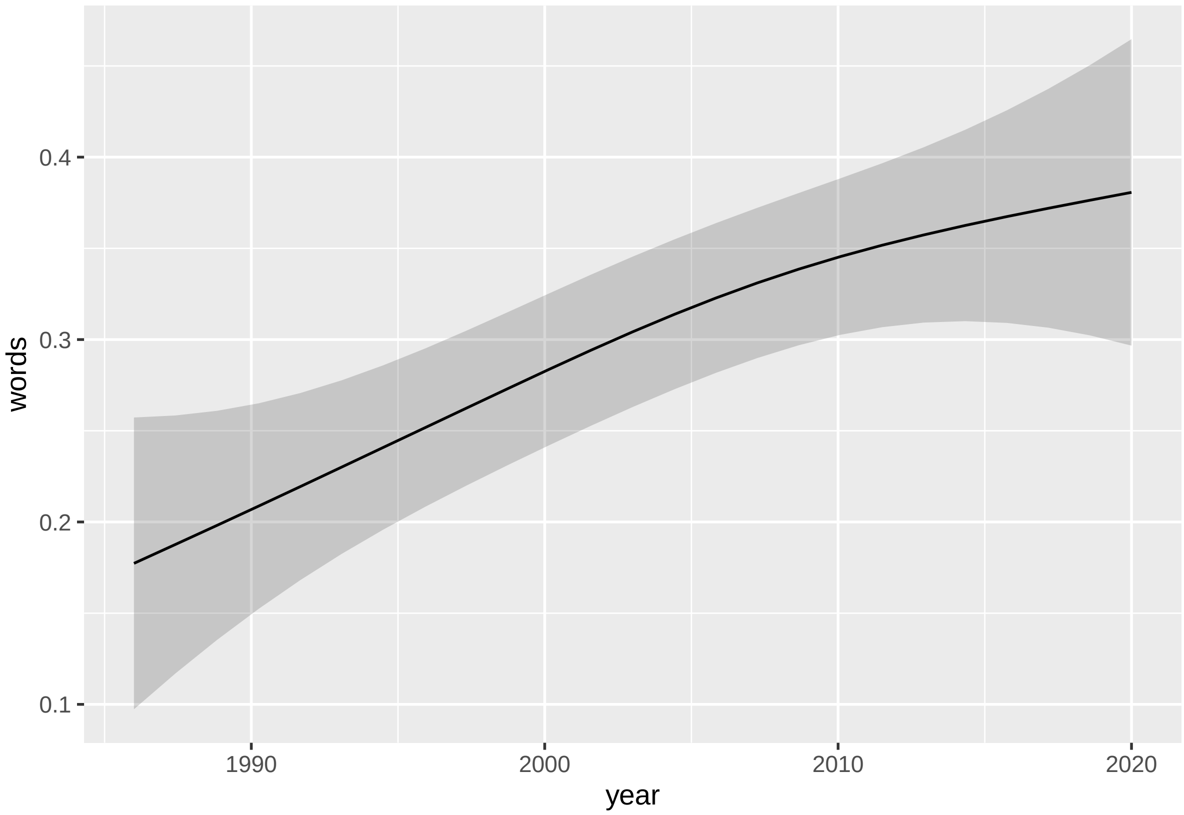
We can also test whether the change over time is non-linear, using a General Additive Model (GAM):

```
gam0 = bam(words ~ s(year), data=dNoOutliers)
# The EDF is a measure of non-linearity
summary(gam0)$edf
```

```
## [1] 1.520483
```

The EDF is slightly above 1, which is an indication of non-linearity. The GAM suggests that the increase in female proportions is slowing down, with the average seeming to plateau around 40%.

```
plot_smooths(gam0,year)
```



However, the GAM does not significantly improve the fit of the model compared to the linear model:

```
lrtest(pm,gam0)
```

```
## Likelihood ratio test
##
## Model 1: words ~ year
## Model 2: words ~ s(year)
##      #Df LogLik      Df  Chisq Pr(>Chisq)
## 1 3.0000 36.652
## 2 3.8797 37.312 0.87972 1.3207     0.2505
```

Therefore we prefer the linear model.

Write some stats for the paper:

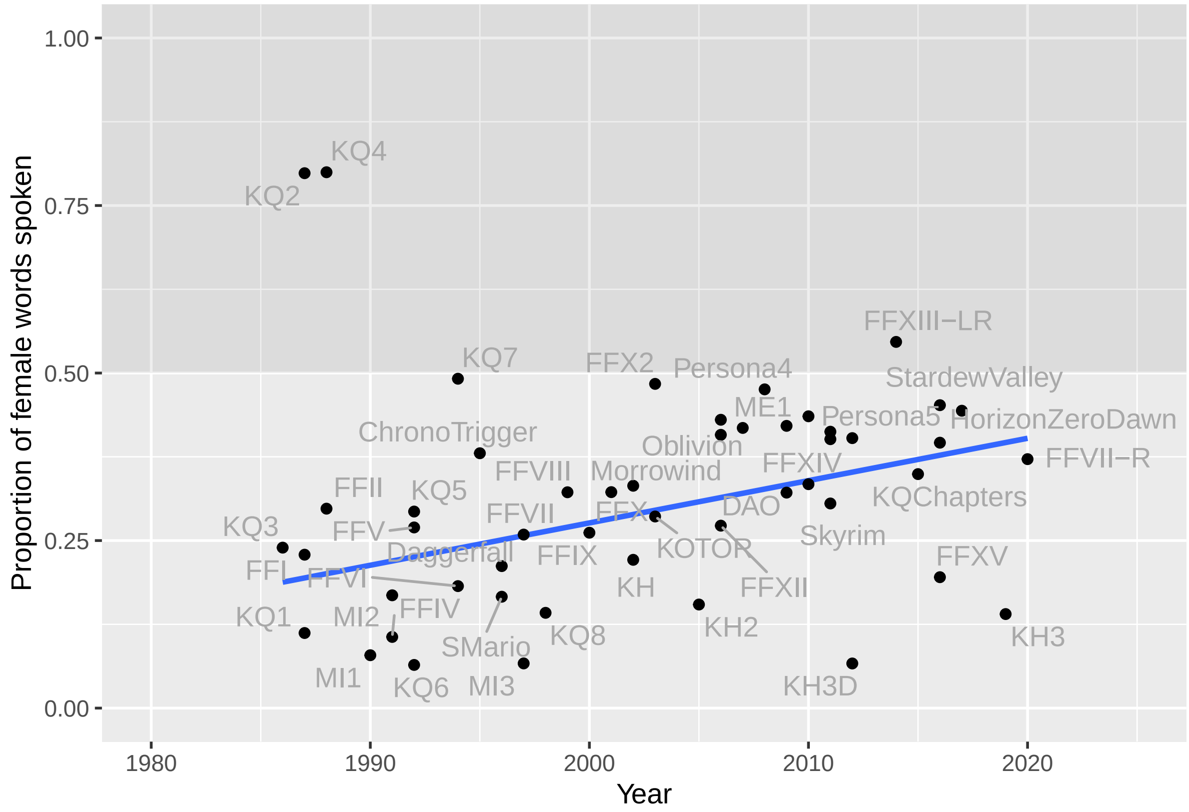
```
propFemaleOverTimeStr = paste0(
  "$r$=",round(propFemaleOverTime$estimate,2),
  " [",round(propFemaleOverTime$conf.int[1],2),",",
    round(propFemaleOverTime$conf.int[2],2),"]",
  ", $n$=",nrow(d),
  ", $p$=",round(propFemaleOverTime$p.value,3))
cat(propFemaleOverTimeStr, file="../results/latexStats/propFemaleOverTime.tex")
cat(incPropFemalePerDecade, file="../results/latexStats/incPropFemalePerDecade.tex")
cat(min(d$year), file="../results/latexStats/earliestYear.tex")
cat(max(d$year), file="../results/latexStats/latestYear.tex")
cat(parity, file="../results/latexStats/yearOfParity.tex")
```

Plot for paper:

```
changeOverTime = ggplot(d,
       aes(x=year,y=words)) +
  annotate("rect", xmin = 0, xmax = 3000,
           ymin = 0.5, ymax = 1.5, alpha = .1) +
#  annotate("rect", xmin = 0, xmax = 3000,
#            ymin = 0.5, ymax = 1.5, alpha = .1,fill="#ec0085") +
#  annotate("rect", xmin = 0, xmax = 3000,
#            ymin = -0.5, ymax = 0.5, alpha = .1,fill="#458fc2") +
  stat_smooth(method=lm, se = F,data = dNoOutliers) +
  geom_point() +
  coord_cartesian(ylim=c(0,1),xlim=c(1980,2025))+
  ylab("Proportion of female words spoken") +
  xlab("Year") +
  geom_text_repel(aes(label=shortName),color="dark gray")
changeOverTime
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: ggrepel: 6 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
pdf('../results/graphs/ChangeOverTime.pdf', width=8,height=6)
changeOverTime
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: ggrepel: 2 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

```
dev.off()
```

```
## pdf
##   2
```

Change in proportions of female characters

There is no significant correlation in the total data:

```
cor.test(d$femCharProp, d$year)
```

```
##
##  Pearson's product-moment correlation
##
## data:  d$femCharProp and d$year
## t = 1.3362, df = 48, p-value = 0.1878
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.09392719  0.44429971
## sample estimates:
##       cor
## 0.1893715
```

But removing one early outlier (King's Quest II) shows that there is a positive trend:

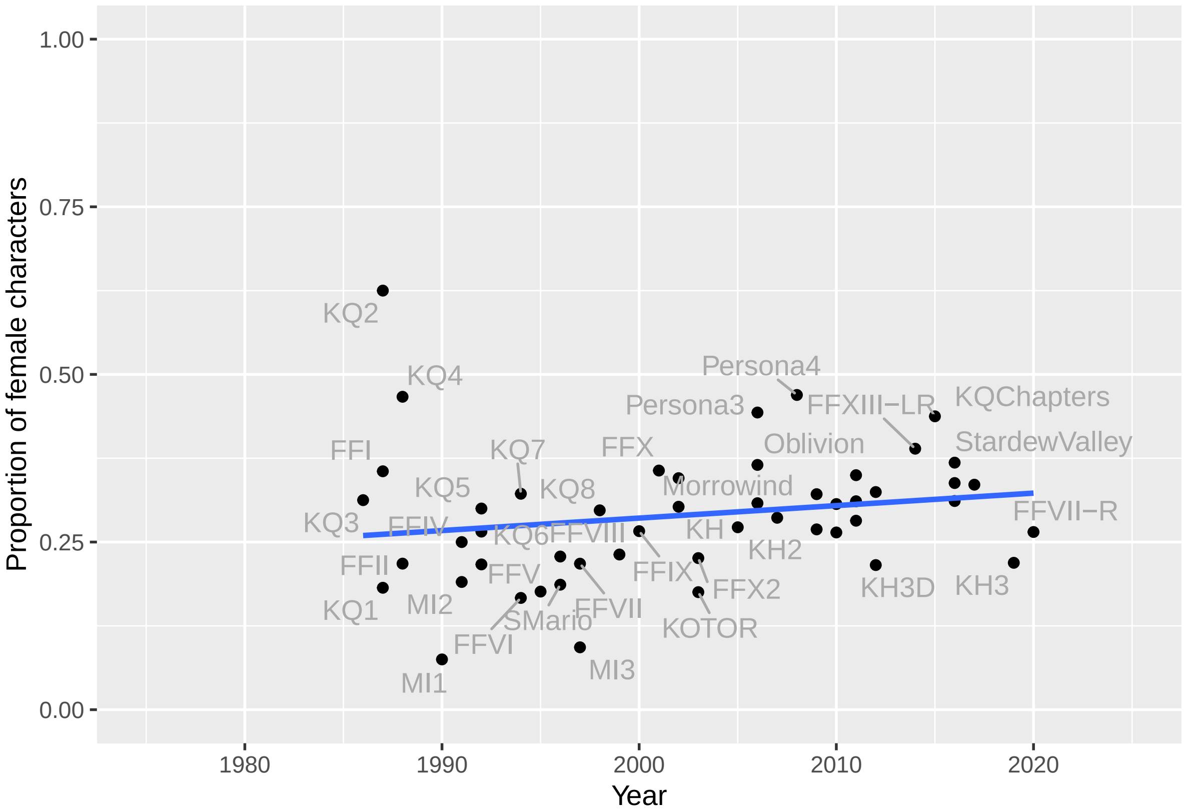
```
dx = d[d$game != "King's Quest II: Romancing the Throne",]
cor.test(dx$femCharProp, dx$year)
```

```
##
##  Pearson's product-moment correlation
##
## data:  dx$femCharProp and dx$year
## t = 2.5814, df = 47, p-value = 0.01302
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.07901041 0.57645514
## sample estimates:
##       cor
## 0.3523781
```

```
ggplot(d,
       aes(x=year,y=femCharProp)) +
  geom_point() +
  stat_smooth(method=lm, se = F) +
  coord_cartesian(ylim=c(0,1),xlim=c(1975,2025))+
  ylab("Proportion of female characters") +
  xlab("Year") +
  geom_text_repel(aes(label=shortName),color="dark gray")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: ggrepel: 15 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```


Proportion of female characters
1.00
0.75
0.50
0.25
0.00
1980
1990
2000
2010
2020
Year
KQ2
KQ4
FFI
KQ3
FFII
KQ1
MI2
MI1
KQ5
FFIV
KQ7
KQ8
KQ6
FFVIII
FFV
FFVI
SMario
FFVII
MI3
FFX
Morrowind
KH
FFIX
FFX2
KOTOR
Persona3
Persona4
Oblivion
KH2
FFXIII−LR
KH3D
KQChapters
StardewValley
FFVII−R
KH3

Assessing gender bias with random baselines

This section analyses the gender balance in the dialogue of the games in the corpus. It begins with an illustration of the statistical methods applied to *Final Fantasy VII*.

The proportion of dialogue by gender is assessed in comparison to two "baselines". These reflect two possible sources of bias in the amount of dialogue spoken by each gender:

- How many characters of each gender there are (i.e. are there more male than female characters).
- How much dialogue each character is given (i.e. are males given more dialogue than females).

Load libraries

```
library(vcd)
library(MASS)
library(ggplot2)
library(ggrepel)
```

Demonstration: Gender differences in Final Fantasy VII

Let's load the data for Final Fantasy VII:

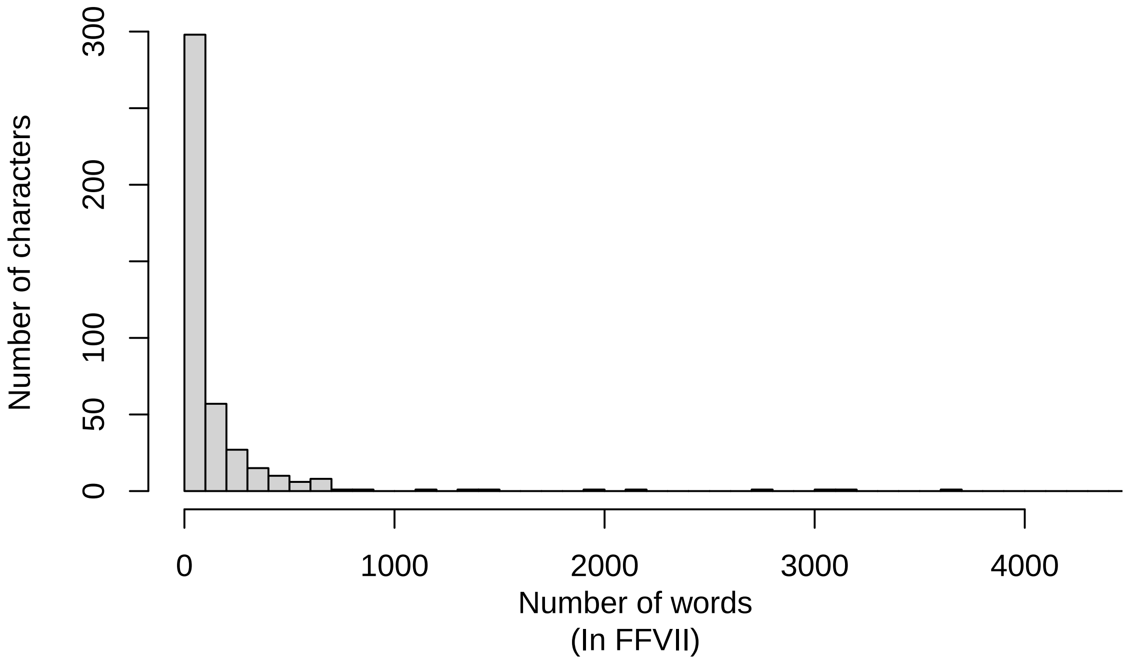
```
d = read.csv("../data/FinalFantasy/FFVII/stats_by_character.csv",stringsAsFactors = F)
cor(d[d$group=="female",]$words,d[d$group=="female",]$lines)
```

```
## [1] 0.9940972
```

Distribution of words by character

What does the distribution of lines per character look like? Here is a plot, showing the number of words of dialogue for each character. The shape of how these values are distributed across the scale is called the "distribution".

```
lengthVar = "words"
# estimate the parameters
distr = d[,lengthVar]
#plot(1:length(distr),sort(distr),xlab="Character",ylab="Words of dialogue")
hist(distr, freq = TRUE, breaks = 100, xlim = c(0, quantile(distr, 0.99)),
     ylab="Number of characters",
     xlab="Number of words\n(In FFVII)", main="")
```



The distribution is very skewed: a small number of characters have a lot of dialogue, and a large number of characters have a small amount of dialogue. For example, 297 characters have less than 100 words of dialogue each, while only 8 characters have over 3,000 words of dialogue each.

We can formally test whether the distribution fits an ideal "normal" (bell-curve), an exponential distribution or a t-distribution:

```
# goodness of fit test
normFit = fitdistr(distr,"normal")
ks.test(distr, "pnorm", normFit$estimate)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  distr
## D = 0.96568, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
expFit = fitdistr(distr, "exponential")
ks.test(distr, "pexp", expFit$estimate)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  distr
## D = 0.34796, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
tFit = fitdistr(distr, "t")
ks.test(distr, "pt", tFit$estimate)
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
```

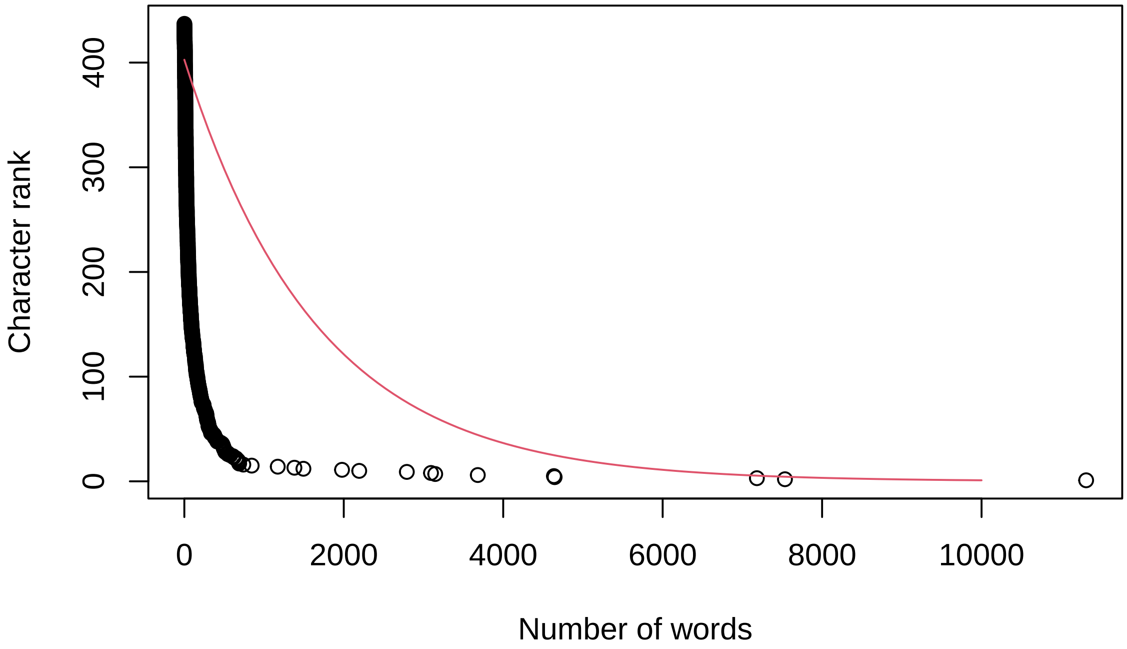
```
## data:  distr
## D = 0.9764, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
# plot a graph
#hist(distr, freq = FALSE, breaks = 100, xlim = c(0, quantile(distr, 0.99)))
#curve(dexp(x, rate = expFit$estimate), from = 0, col = "red", add = TRUE)
```

We see that the dialogue by character distribution does not fit any of these ideal distributions. This means that using standard tests such as the t-test is not appropriate in order to compare the distributions (e.g. for testing if the average number of words given to male characters is higher than the average number of words given to female characters).

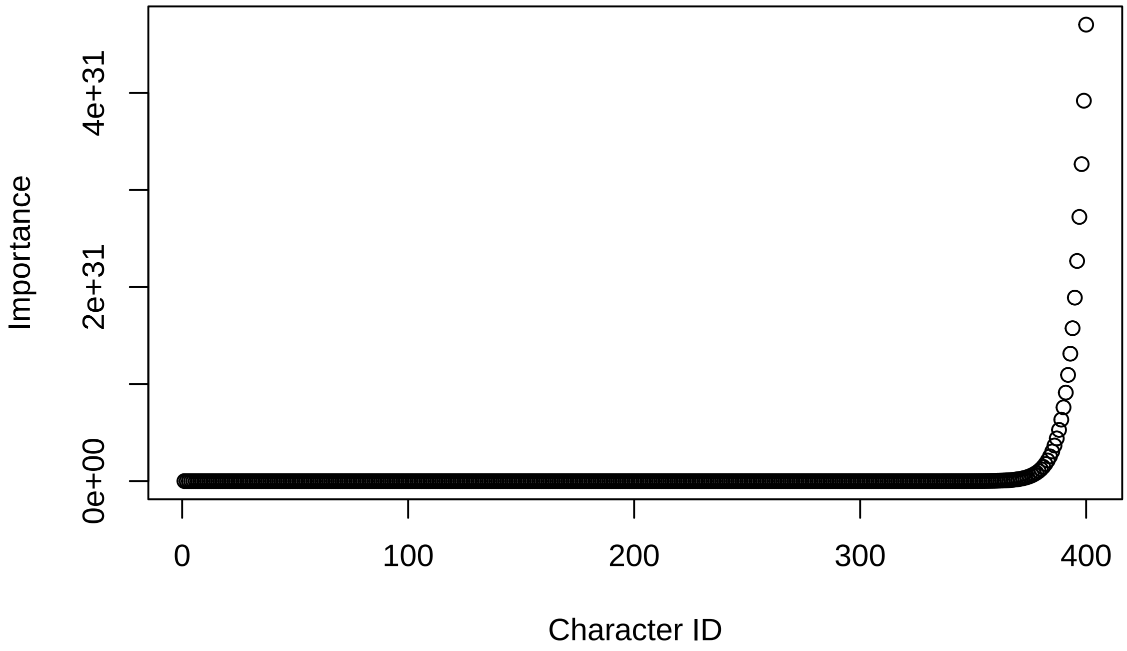
We can see the departure from an exponential curve by looking at the number of words spoken by a character compared to their rank (black circles): there is a very sharp decrease in characters, then a smaller set of 'main' characters. The turning point is a kind of "elbow" shape. The plot below includes a true exponential line in red for reference:

```
plot(1:nrow(d)~sort(d$words,decreasing = T),
     xlab="Number of words",ylab="Character rank")
# Exponential line
ys = seq(0,10000,length.out = 100)
points(ys,rev(1.0006^(ys)),type='l',col=2)
```



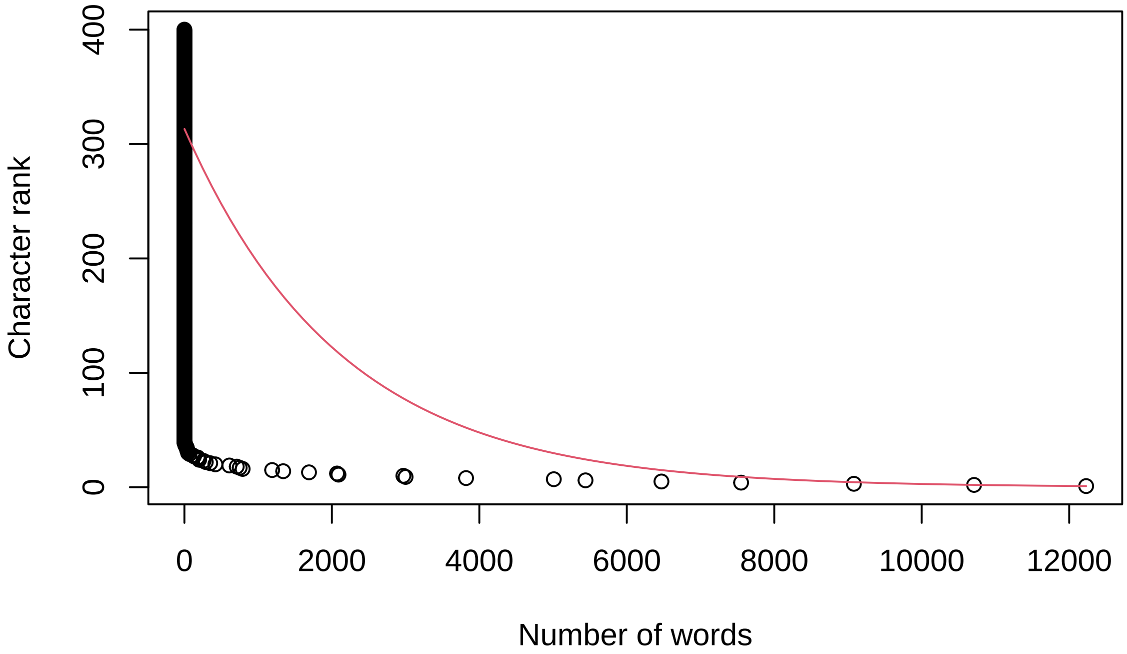
The "elbow" distribution we see in the data can be generated using the following method: Assume that we have 400 characters, and that the distribution of their importance to the plot is exponential (a few characters are much more likely to be chosen to speak than the majority)

```
nChar = 400
# Exponential character importance
charImportance = 1.2^(1:nChar)
plot(charImportance,xlab="Character ID",ylab="Importance")
```



Now we can simulate a conversation: We pick one character at random, but weighted in proportion to their importance (we're more likely to choose a more important character). Now pick one other character (that is not the first character) in the same way. They each say 10 words. We run this simple simulation many times, and show that it generates the same kind of "elbow" (black dots). A true exponential line is shown in red for comparison. With comparable settings for FFVII, it also generates roughly the same number of words that the most prolific character speaks.

```
simulateConversation = function(){
  chars = 1:nChar
  char1 = sample(chars,1,prob = charImportance)
  char2 = sample(chars[chars!=char1],1,prob = charImportance[chars!=char1])
  # 10 words each
  10* ((1:nChar) %in% c(char1,char2))
}

# 8000 lines in FFVII, which is 4000 pairs of lines
res = replicate(4000,simulateConversation())
distrSim = rowSums(res)

plot(1:length(distrSim)~sort(distrSim,decreasing = T),
     xlab="Number of words",ylab="Character rank")
# Exponential line
ys = seq(1,max(distrSim),length.out = 100)
points(ys,rev(1.00047^(ys)),type='l',col=2)
```



So, the distribution of words by characters may be a product of conversations being *interactive*: at least two speakers are required to have a conversation.

Difference in words per gender

Below we calculate the total number of words for male and female characters. There are other gender groups, but we'll focus just on the comparison between male and female.

```
dx = d[d$group %in% c("male","female"),]
totalWords = tapply(dx[,lengthVar],dx$group,sum)
cbind(words=totalWords,proprtion=totalWords/sum(totalWords))
```

```
##        words proprtion
## female 25102 0.2589437
## male   71838 0.7410563
```

Total number of male female characters.

```
totalCharacters = table(dx$group)
cbind(characters=totalCharacters, proportion = prop.table(totalCharacters))
```

```
##        characters proportion
## female         81  0.2177419
## male          291  0.7822581
```

It's clear that the proportion of male dialogue is to some extent affected by the proportion of male characters. In order to tell whether there is a bias for men to speak more than women independently of the number of male and female characters, we need to compare the true difference in the data to a "baseline". Baselines are a calculation of the range of differences we would expect in particular conditions, such as there being no bias in dialogue length by gender.

Therefore, we calculate the true difference in total number of words for male and female characters in our data. A positive number indicates more words for males, expressed in number of words:

```
true_diffInWords = diff(totalWords)
true_diffInWords
```

```
##  male
## 46736
```

Baseline 1: Randomly assigned gender

In order to contextualise the difference between the amount of male and female dialogue, we need to compare it to a baseline. We use two baselines in this study: In the first baseline, we hold fixed which lines belong to which characters, and randomly assign them a gender (creating roughly 50% female characters). In the second, we randomly shuffle the genders (preserving the proportion of characters). This is explained as follows:

We can simulate what would happen if the gender of each character was assigned randomly, with an even chance of each character being male or female. This can lead to a different number of male and female characters compared to the true game. A typical simulation would have roughly 50% male and 50% female. We know that this baseline is **unbiased**, at least in the sense that there is an equal probability of any character being male or female.

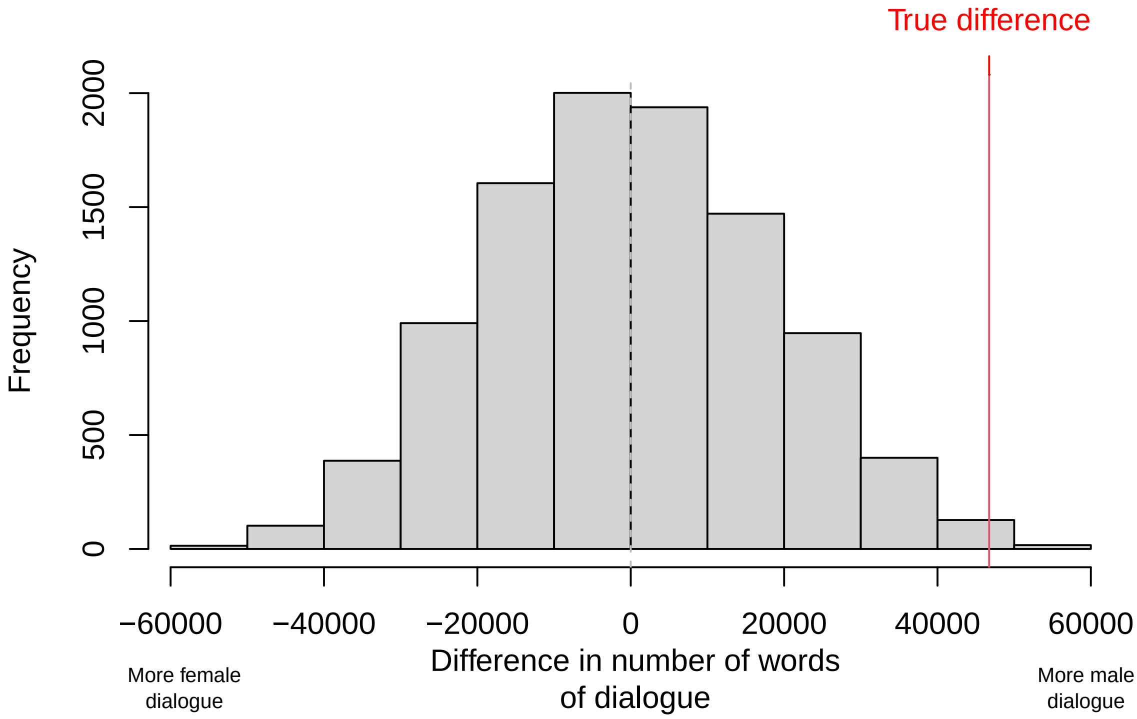
In the simulation, we assign gender randomly then work out the difference in the total number of words between male and female characters. We run this simulation 10,000 times to get lots of values for this measure (a "distribution").

```
randomGender = function(words,groups){
  diff(tapply(
        words,
        sample(unique(groups),length(words),replace = T),
        sum))
}
set.seed(451)
random_diffInWords = replicate(10000,
                         randomGender(dx[,lengthVar],dx$group))
```

So we now have two types of data: the true difference between the amount of male and female dialogue that we observed in the data. We'll call this the "true gender bias". And we also have lots of estimates of the difference between the amount of male and female dialogue if the gender was assigned randomly. This forms our baseline, and we'll call this the "random distribution".

We can now visualize these values: the random distribution is visualised below as a histogram, and the true gender bias is marked with a red line. If the true gender bias in dialogue is not different from the random distribution, then the true bias should lie in the middle of the random distribution. That is, it could have plausibly been generated by randomly assigning genders. However, in the graph below we see that the true gender bias lies at one extreme of the random distribution. This indicates that it is unlikely that the gender bias we see in the real data was generated by randomly assigning genders. That is, the true gender bias is very different from the baseline.

```
hist.compareDist = function(perm,trueVal){
  pmin = min(c(trueVal,perm,0))
  pmax = max(c(trueVal,perm,0))
  hist(perm, xlim=c(pmin,pmax),
       xlab="Difference in number of words\nof dialogue",
       main="")
  abline(v=0,col='gray',lty=2)
  axis(1,c(pmin,pmax),
       c("More female\ndialogue","More male\ndialogue"),
       tick=F,line=2,cex.axis=0.7)
  abline(v=trueVal,col=2)
  axis(3,trueVal,"True difference",col.axis="red",col.ticks="red")
}
hist.compareDist(random_diffInWords,true_diffInWords)
```



We can quantify the difference between the true bias and the random distribution. The first measure is the z-score, which indicates how far away the true difference is from the mean, expressed in the number of standard deviations. The second measure is a p-value, which indicates the proportion of simulations that resulted in a more extreme measure of difference than the true value.

```
zstats = function(true,dist){
  zscore = (true-mean(dist)) / sd(dist)
  pvalue = 1/length(dist)
  numAgainst = sum(true < dist)
  if(numAgainst>0){
    pvalue = numAgainst/ length(dist)
  }
  return(c(zscore=zscore,p=pvalue))
}
zstats(true_diffInWords,random_diffInWords)
```

```
## zscore.male           p
##    2.531894    0.003500
```

The p-value is less than 0.05, suggesting that there is less than 5% chance of observing the true bias in a simulation where gender is assigned randomly. That is, the true gender bias is unlikely to have been generated by an unbiased process. Put another way, the difference between male and female dialogue is significant in comparison to this baseline.

Baseline 2: Permuted gender

We can also specify a different baseline that takes into account the proportion of male and female characters. This can be done by randomly swapping the gender of characters. The difference from the first baseline is as follows: Imagine some real data where the characters were composed of 30% female characters and 70% male characters. The first baseline would change that proportion to roughly 50%/50%. In contrast, the simulation for the second baseline preserves the same proportion (30% female, 70% male), but with a different assignment of which characters are male and which are female.

There's another way to think about this second baseline that is an equivalent interpretation. Although the implementation below randomly permutes the gender values, it's effectively equivalent to permuting the values of the dialogue. That is, the process effectively randomly swaps all of one character's lines for all of another character's lines. For example, all of (main protagonist) Cloud's lines are now spoken by (party member) Aerith, and all of Aerith's lines are now spoken by (minor character) "Shopkeeper (Items Sector 7)".

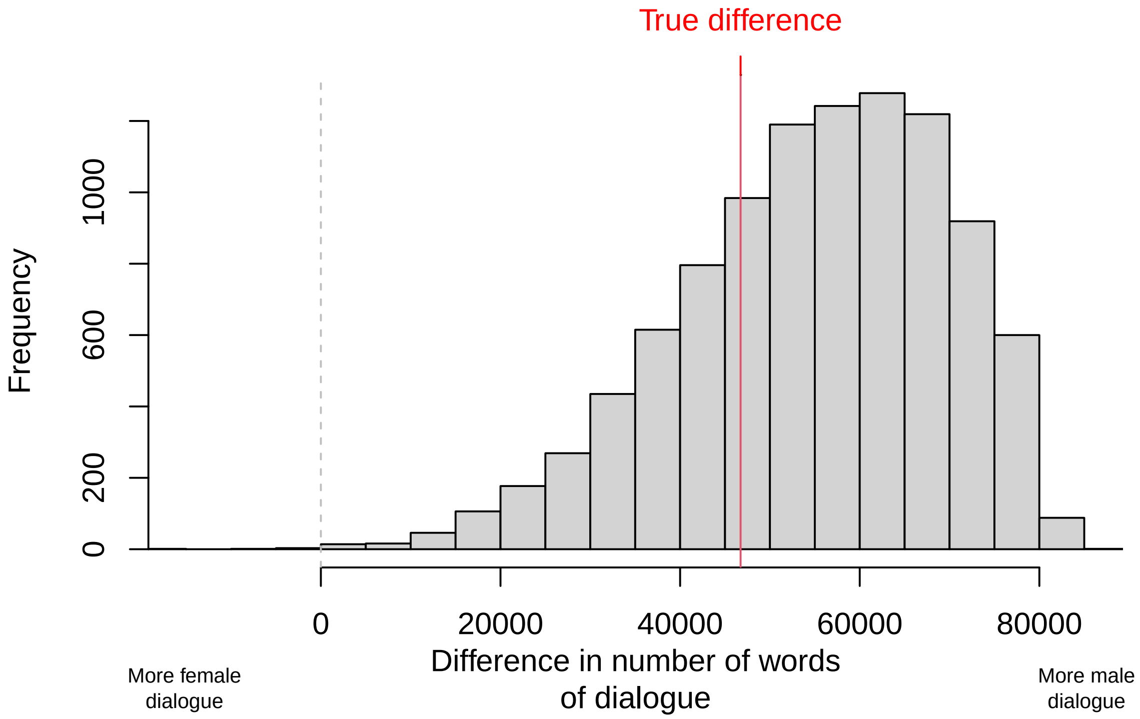
If there was no gender bias in the amount of dialogue written for each character, then this random swapping would not produce a big change in the calculation of the difference between genders. So the baseline is unbiased in the sense that we know that there's no bias in the amount of dialogue given to any particular character.

The function below works out the difference in the sum of words between the two gender groups, but where the assignment of characters to gender groups has been randomly permuted (sampled). This preserves the same number of male and female characters as in the true data.

```
permuteGender = function(words,group){
  diff(tapply(words,sample(group),sum))
}
set.seed(451)
permuted_diffInWords = replicate(10000,
                          permuteGender(dx[,lengthVar],dx$group))
```

We'll call this second baseline the "permuted distribution". We can now visualise the permuted distribution from these second baseline simulations, and compare it to the true gender bias This time, we see that the true gender bias lies in the middle of the distribution:

```
hist.compareDist(permuted_diffInWords,true_diffInWords)
```



And the stats also suggest that the true difference is reasonably likely to be generated by the permuted simulation. That is, it's plausible that the true data was generated by a process that had no bias for the amount of dialogue given to a character based on their gender.

```
zstats(true_diffInWords,permuted_diffInWords)
```

```
## zscore.male          p
##  -0.5441816  0.7197000
```

Summary

We discovered a few things about Final Fantasy VII:

- The distribution of dialogue by character does not approximate common distributions, preventing straightforward statistical tests.
- The difference in the number of words of dialogue between male and female characters is biased. There are more words spoken by male characters compared to a game where the gender of each character was randomly determined.
- However, there is no dialogue bias in Final Fantasy VII. That is, the proportion of dialogue reflects the proportion of male and female characters, and there's no evidence that the average female character is given fewer lines than the average male character.

In principle, the two types of test are independent: a game might have a small or large proportion of female characters, and those characters might have more or less to say than male characters.

Applying the baselines to all games

We can now apply the methods above to all the games in the corpus. These calculations were run offline in the script `compareToBaselines.R`.

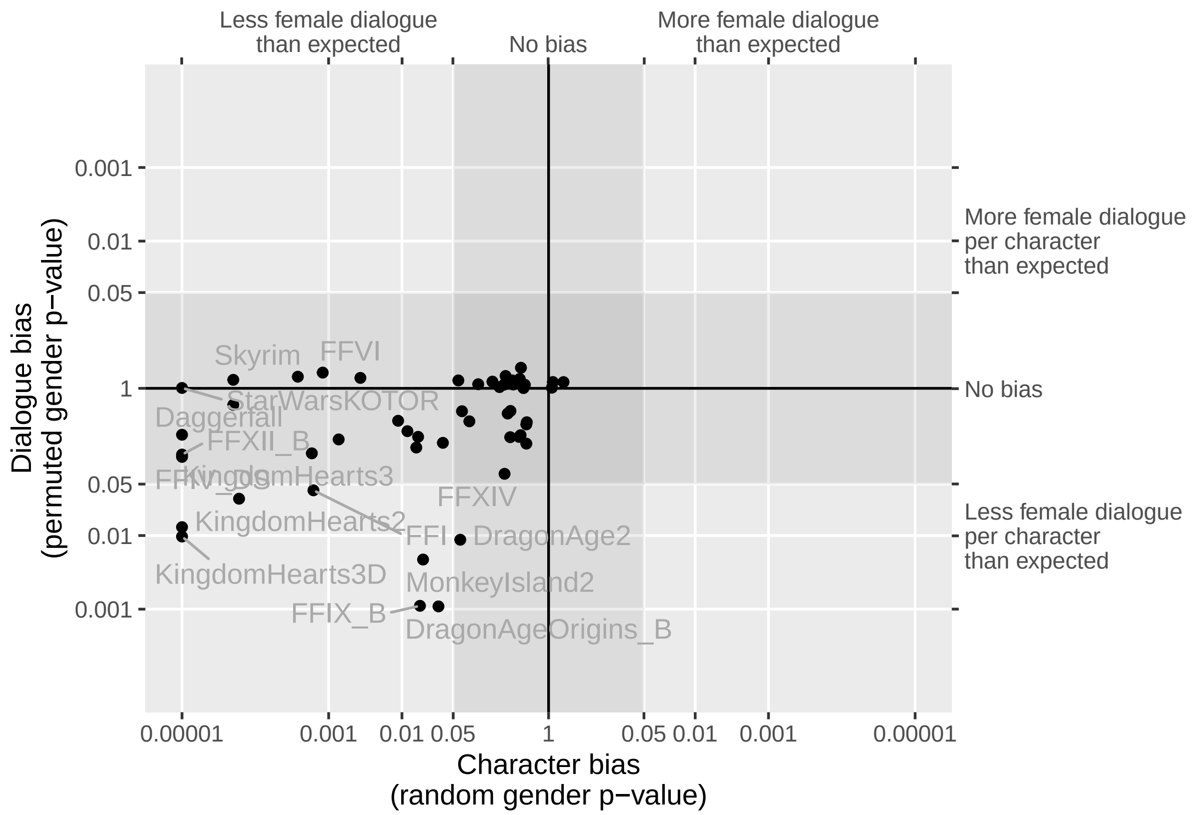
Some games have much larger scripts than others. So for the test that compares the entire corpus as a whole, the length of dialogue was transformed to words-per-thousand within a game.

```
all = read.csv("../results/compareToBaseline.csv",stringsAsFactors = F)
all = all[!is.na(all$p.perm),]
all = all[all$game!="Test",]
all = all[!all$alternativeMeasure,]
```

Create a bias plot for the publication:

```
all$p.random.log = log10(all$p.random)
all$p.perm.log = log10(all$p.perm)

all$p.random.log[all$z.random<0] = -all$p.random.log[all$z.random<0]
all$p.perm.log[all$z.perm<0] = -all$p.perm.log[all$z.perm<0]

threshold = log10(0.05)

options(scipen=999)
xs = c(0.05,0.01,0.001,0.00001)
xs2 = c(rev(xs),1,xs)
ls = c(log10(rev(xs)),0,-log10(xs))

gx = ggplot(all, aes(x=p.random.log,y=p.perm.log)) +
  annotate("rect", xmin = -1000, xmax = 1000, ymin = -threshold, ymax = threshold, alpha = .1) +
  annotate("rect", xmin = -threshold, xmax = threshold, ymin = -1000, ymax = 1000, alpha = .1) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  geom_text_repel(aes(label=shortName),color="dark gray",force = 1) +
  coord_cartesian(ylim=c(-4,4),xlim=c(-5,5)) +
  scale_x_continuous(breaks=ls,labels=xs2,
                     sec.axis = sec_axis(~.*1,
                        breaks = ls,
                        labels=c("","Less female dialogue\nthan expected","","",
                                 "No bias",
                                 "","",
                                 "More female dialogue\nthan expected",""))) +
  scale_y_continuous(breaks=ls,labels=xs2,
                     sec.axis = sec_axis(~.*1,
                        breaks = ls,
                        labels=c("","","Less female dialogue\nper character\nthan expected","",
                                 "No bias","",
                                 "More female dialogue\nper character\nthan expected","",""))) +
  theme(panel.grid.minor = element_blank()) +
  xlab("Character bias\n(random gender p-value)") +
  ylab("Dialogue bias\n(permuted gender p-value)")
gx
```

```
## Warning: ggrepel: 35 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



Write to file

```
pdf("../results/graphs/CompareToBaseline.pdf",width=9.5,height=7)
gx
```

```
## Warning: ggrepel: 28 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

```
dev.off()
```

```
## pdf
##   2
```

The graph above shows the p-values for the random baseline compared to the permuted baseline. Values nearer the center are not significantly different from the baseline. The gray rectangles show the threshold of p < 0.05.

Full results for the comparison to the random baseline:

```
knitr::kable(all[,c("game","z.random","p.random")])
```

	game	z.random	p.random
1	Chrono Trigger	0.4712166	0.49771
2	Dragon Age 2	1.0980156	0.06246
5	Dragon Age: Origins	2.8100350	0.03148
6	The Elder Scrolls II: Daggerfall	6.4962123	0.00001
7	The Elder Scrolls III: Morrowind	0.6609665	0.50423
8	The Elder Scrolls IV: Oblivion	1.5651459	0.05877
9	The Elder Scrolls V: Skyrim	3.2730719	0.00038
10	Final Fantasy	2.5829984	0.00062
11	Final Fantasy II	0.7891011	0.39550
14	Final Fantasy IV	3.3840605	0.00006
16	Final Fantasy IX	3.6726997	0.01761
17	Final Fantasy V	0.8898893	0.29979

	game	z.random	p.random
18	Final Fantasy VI	2.9433827	0.00083
21	Final Fantasy VII	2.5199998	0.00271
22	Final Fantasy VII Remake	1.2575180	0.10966
23	Final Fantasy VIII	0.6822423	0.41525
25	Final Fantasy X	2.1648299	0.00887
26	Final Fantasy X-2	0.1337843	0.45468
28	Final Fantasy XII	6.1437520	0.00001
29	Final Fantasy XIII	0.4819298	0.31968
30	Final Fantasy XIII-2	0.6035257	0.25938
31	Lightning Returns: Final Fantasy XIII	-0.2500463	0.62486
32	Final Fantasy XIV	0.9227224	0.25101
33	Final Fantasy XV	2.0950667	0.01570
34	Horizon Zero Dawn	0.4030046	0.40493
36	Kingdom Hearts	2.7962376	0.00059
37	Kingdom Hearts II	3.4460522	0.00001
38	Kingdom Hearts III	3.3153632	0.00001
39	Kingdom Hearts 3D: Dream Drop Distance	3.4228924	0.00001
40	King's Quest I: Quest for the Crown	2.0820536	0.01660
41	King's Quest II: Romancing the Throne	-1.2977746	0.87385
42	King's Quest III: To Heir Is Human	1.4005520	0.08311
43	King's Quest IV: The Perils of Rosella	-1.3980067	0.90203
44	King's Quest V	1.5237709	0.06591
45	King's Quest VI	1.3731348	0.00137
46	King's Quest VII: The Princeless Bride	0.0578734	0.47490
47	King's Quest VIII	1.4145117	0.01187
48	King's Quest Chapters	0.8082777	0.30192
50	Mass Effect	0.7143946	0.24692
51	Mass Effect 2	0.8067654	0.21425
54	Mass Effect 3	0.3878197	0.49747
56	Monkey Island 2: LeChuck's Revenge	2.5432044	0.01942
57	The Curse of Monkey Island	2.0836895	0.00005
58	The Secret of Monkey Island	2.4678511	0.00005
59	Persona 3	0.6137222	0.27697
60	Persona 4	0.2201780	0.42006
62	Persona 5	0.9781966	0.17144
63	Stardew Valley	0.4491905	0.32962
64	Star Wars: Knights of the Old Republic	3.5698348	0.00001
65	Super Mario RPG: Legend of the Seven Stars	1.2998554	0.03616

Full results for the comparison to the permuted baseline:

```
knitr::kable(all[,c("game","z.perm","p.perm")])
```

	game	z.perm	p.perm
1	Chrono Trigger	0.6102605	0.17736
2	Dragon Age 2	1.6960572	0.00877
5	Dragon Age: Origins	3.5510906	0.00109
6	The Elder Scrolls II: Daggerfall	0.7457174	0.23421
7	The Elder Scrolls III: Morrowind	0.7005772	0.34632
8	The Elder Scrolls IV: Oblivion	-0.7765375	0.78045
9	The Elder Scrolls V: Skyrim	-0.4950374	0.69546
10	Final Fantasy	1.4448200	0.04117
11	Final Fantasy II	0.9485726	0.21896
14	Final Fantasy IV	1.6481142	0.03165
16	Final Fantasy IX	4.3839283	0.00111
17	Final Fantasy V	1.0615415	0.21703

	game	z.perm	p.perm
18	Final Fantasy VI	-0.2238571	0.61135
21	Final Fantasy VII	-0.5420979	0.72154
22	Final Fantasy VII Remake	-1.2249782	0.87908
23	Final Fantasy VIII	0.8096734	0.23098
25	Final Fantasy X	0.4530733	0.36258
26	Final Fantasy X-2	-2.6913868	0.99228
28	Final Fantasy XII	1.1515818	0.12595
29	Final Fantasy XIII	-0.7842501	0.77992
30	Final Fantasy XIII-2	-0.4611162	0.67899
31	Lightning Returns: Final Fantasy XIII	-0.8750701	0.82245
32	Final Fantasy XIV	1.6103386	0.06900
33	Final Fantasy XV	1.1291897	0.15687
34	Horizon Zero Dawn	-0.8638365	0.74498
36	Kingdom Hearts	1.0852511	0.13089
37	Kingdom Hearts II	1.4546342	0.01302
38	Kingdom Hearts III	0.9764778	0.11716
39	Kingdom Hearts 3D: Dream Drop Distance	1.7027103	0.00970
40	King's Quest I: Quest for the Crown	0.7852188	0.21920
41	King's Quest II: Romancing the Throne	-1.1467688	0.82143
42	King's Quest III: To Heir Is Human	0.5537754	0.35642
43	King's Quest IV: The Perils of Rosella	-1.8757304	0.98008
44	King's Quest V	0.0712877	0.48851
45	King's Quest VI	0.7271370	0.20225
46	King's Quest VII: The Princeless Bride	-1.3483466	0.89441
47	King's Quest VIII	0.7016147	0.26181
48	King's Quest Chapters	0.5324388	0.49298
50	Mass Effect	-1.3162577	0.89131
51	Mass Effect 2	-1.8439488	0.96110
54	Mass Effect 3	0.4105362	0.32394
56	Monkey Island 2: LeChuck's Revenge	5.3020303	0.00472
57	The Curse of Monkey Island	0.2320127	0.59692
58	The Secret of Monkey Island	-0.0479110	0.76657
59	Persona 3	0.1318010	0.45482
60	Persona 4	-0.0602077	0.52554
62	Persona 5	-0.8941771	0.81080
63	Stardew Valley	-1.2221531	0.88629
64	Star Wars: Knights of the Old Republic	-2.6013485	0.98781
65	Super Mario RPG: Legend of the Seven Stars	1.6607465	0.18181

Mini graph for publication:

```r
xs = c(0.05,0.001,0.00001)
xs2 = c(rev(xs),1,xs)
ls = c(log10(rev(xs)),0,-log10(xs))
gxMini = ggplot(all, aes(x=p.random.log,y=p.perm.log)) +
  annotate("rect", xmin = -1000, xmax = 1000, ymin = -threshold, ymax = threshold, alpha = .1) +
  annotate("rect", xmin = -threshold, xmax = threshold, ymin = -1000, ymax = 1000, alpha = .1) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_vline(xintercept = 0) +
  coord_cartesian(ylim=c(-4,4),xlim=c(-5,5)) +
  scale_x_continuous(breaks=ls,labels=xs2,
                     sec.axis = sec_axis(~.*1,
                        breaks = ls,
                        labels=c("","Less female\ndialogue\nthan expected","",
                                 "No bias",
                                 "",
                                 "More female\ndialogue\nthan expected",""))) +
  scale_y_continuous(breaks=ls,labels=xs2,
                     sec.axis = sec_axis(~.*1,
                        breaks = ls,
                        labels=c("","Less female\ndialogue\nper character\nthan expected","",
                                 "No bias","",
                                 "More female\ndialogue\nper character\nthan expected", ""))) +
  theme(panel.grid.minor = element_blank()) +#,panel.grid.major = element_blank(), axis.ticks = elem
  xlab("Character bias\n(random gender p-value)") +
  ylab("Dialogue bias\n(permuted gender p-value)")
pdf("../results/graphs/CompareToBaseline_Mini.pdf",height=4,width=5)
gxMini
dev.off()
```

```
## pdf
##   2
```

Big plot for paper

```
# pdf("../results/graphs/Big3.pdf",width=10,height=6)
# ggarrange(ggarrange(gxMini,
#                     wordsVCharacters+ geom_point(),
#                     nrow = 2, labels = c("A", "B")),
#           changeOverTime,
#           ncol = 2,
#           labels = "C",widths = c(1,1.5))
# dev.off()

pdf("../results/graphs/Big2.pdf",width=8,height=4)
ggarrange(gxMini,
          wordsVCharacters+ geom_point(),
          ncol = 2, labels = c("A", "B"),widths=c(1.2,1))
dev.off()
```

```
## pdf
##   2
```

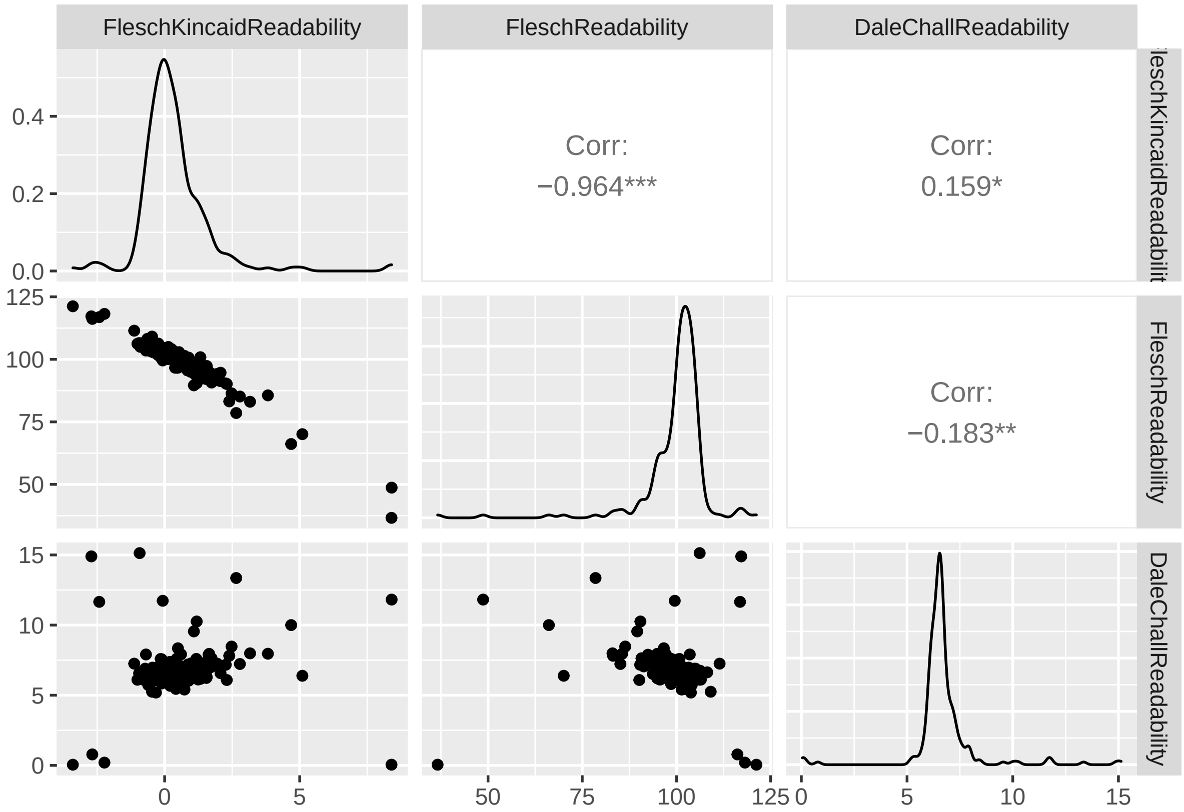
Readability

There are multiple measures of readability calculated by Textatistic:

- Flesch score
- Flesch Kincaid score
- Dale-Chall score

The Flesch and Dale-Chall measures are not strongly correlated with each other:

```
readMeasures = c("FleschKincaidReadability","FleschReadability","DaleChallReadability")
ggpairs(stats[,readMeasures])
```



```
getReadabilityTestText = function(readability,rt){
  p = rt$p.value
  if(p<0.0001){
    p  = "< 0.0001"
  } else{
    p = round(p,3)
  }
  paste0("game-level mean readability for male characters = ",
         round(mean(readability[1,],na.rm=T),2),
         ", sd = ",
         round(sd(readability[1,],na.rm=T),2),
         ", for female characters = ",
         round(mean(readability[2,],na.rm=T),2),
         ", sd = ",
         round(sd(readability[2,],na.rm=T),2),
         ", paired t-test t = ",
         round(rt$statistic,2),
         ", n = ",
         sum(!is.na(readability[1,])),
         ", p ",p
  )
}

games = unique(stats$folder)
grpPerGame =tapply(stats$group,stats$folder,length)
games = games[games %in% names(grpPerGame[grpPerGame>1])]

readability.DC = sapply(games, function(g){
  c(stats[stats$folder==g & stats$group=="male",]$DaleChallReadability,
    stats[stats$folder==g & stats$group=="female",]$DaleChallReadability)})
readability.DC.t = t.test(readability.DC[1,], readability.DC[2,], paired = T)
readability.DC.text = getReadabilityTestText(readability.DC,readability.DC.t)
cat(readability.DC.text,file="../results/latexStats/readability-DC-TTest.tex")

readability.F = sapply(games, function(g){
  c(stats[stats$folder==g & stats$group=="male",]$FleschReadability,
    stats[stats$folder==g & stats$group=="female",]$FleschReadability)})
readability.F.t = t.test(readability.F[1,], readability.F[2,], paired = T)
readability.F.text = getReadabilityTestText(readability.F,readability.F.t)
cat(readability.F.text,file="../results/latexStats/readability-F-TTest.tex")

readability.FK = sapply(games, function(g){
  c(stats[stats$folder==g & stats$group=="male",]$FleschKincaidReadability,
    stats[stats$folder==g & stats$group=="female",]$FleschKincaidReadability)})
readability.FK.t = t.test(readability.FK[1,], readability.FK[2,], paired = T)
readability.FK.text = getReadabilityTestText(readability.FK,readability.FK.t)
cat(readability.FK.text,file="../results/latexStats/readability-FK-TTest.tex")
```

The statistical results are as follows:

Dale-Chall: game-level mean readability for male characters = 6.58, sd = 0.37, for female characters = 6.42, sd = 0.41, paired t-test t = 4.58, n = 50, p < 0.0001

Flesch: game-level mean readability for male characters = 101.5, sd = 3.39, for female characters = 101.34, sd = 3.81, paired t-test t = 0.4, n = 50, p 0.692

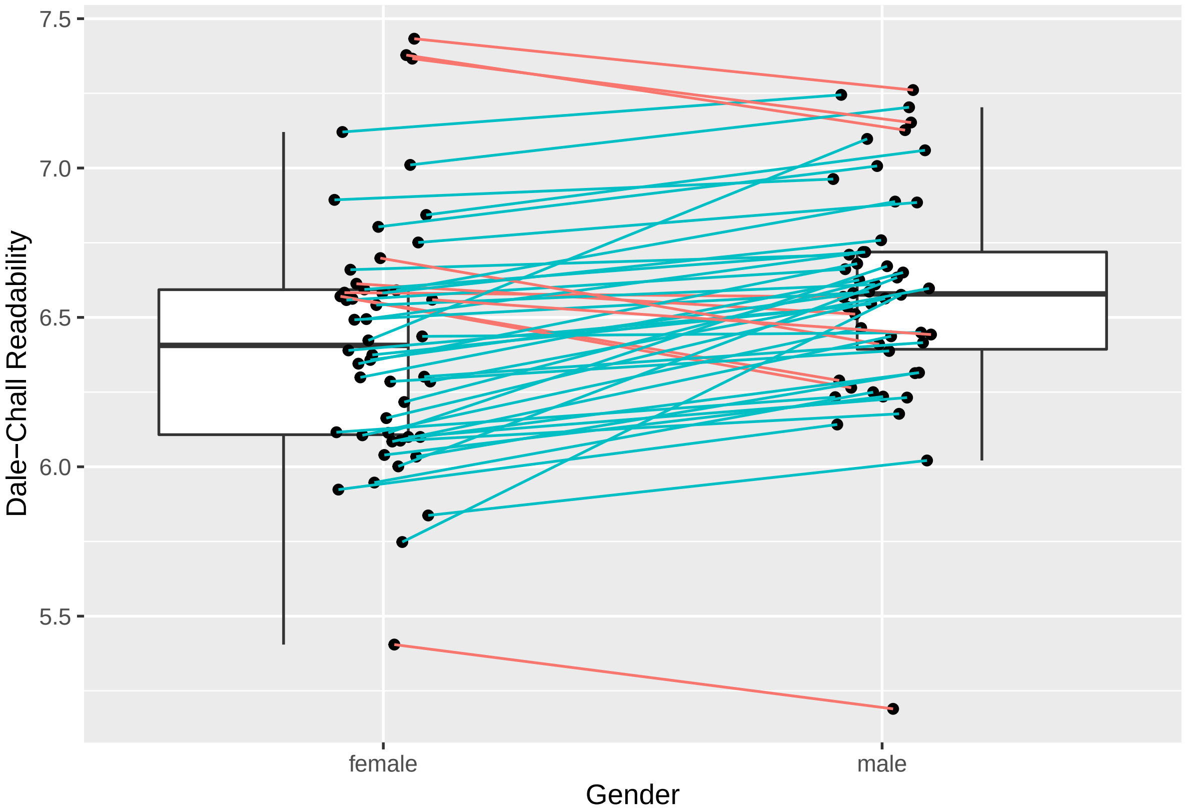
Flesch-Kincaid: game-level mean readability for male characters = 0.16, sd = 0.68, for female characters = 0.18, sd = 0.77, paired t-test t = -0.36, n = 50, p 0.719

For the Dale-Chall scores, male dialogue has significantly higher values than female dialogue. This suggests that the required reading grade for male text is higher, or in other words male dialogue includes a smaller proportion of high-frequency words. The effect size is roughly the same as for the difference in reading ease between male and female academic journal publications (Hengel, 2022, table 3).

Below, we plot Dale-Chall scores, with values from the same game being connected by a line. The lines are coloured by blue = male > female, red = female > male :

```
stats = stats[order(stats$folder,stats$group),]
dir = tapply(stats[stats$group %in% c("male","female"),]$DaleChallReadability,
             stats[stats$group %in% c("male","female"),]$folder,
             function(X){X[1]<X[2]})
stats$dir = dir[stats$folder]
dcBox = stats[stats$group %in% c("male","female"),] %>%
  ggplot(aes(x=group,y=DaleChallReadability)) +
  geom_boxplot(outlier.alpha = 0, width = 0.5,
               position = position_nudge(x=c(-0.2,0.2))) +
  geom_point(aes(group=folder), position = position_dodge(0.2)) +
  geom_line(aes(group=folder,colour=dir),position = position_dodge(0.2))+
  xlab("Gender") +
  ylab("Dale-Chall Readability")+
  theme(legend.position = "none")
```

```
dcBox
```



```
pdf("../results/graphs/Readability.pdf",width=6,height=4)
dcBox
dev.off()
```

```
## pdf
##   2
```

However, the Flesch measures are not significant. In contrast, in Hengel (2022, table 3), the Flesch measures have stronger effect sizes than the Dale-Chall scores. Taken together with the lack of correlation between the measures, this casts some doubt on the robustness of the results for readability.

Discussion

This report demonstrated several things:

- There is about twice as much male dialogue than female dialogue in video games.
- There is high agreement between different measures of the amount of dialogue (number of words, lines, syllables).
- The proportion of female words is increasing over time.
- There is a significant correlation between the proportion of female characters and the proportion of female dialogue.
- The average male character does not say significantly more than the average female character.
- There is considerable variation between games on the gender difference in number of words per sentence.
- There are some differences in readability between male and female dialogue, though the results are not robust over different measures of readability.

References

Hengel, E. (2022) Publishing while female: Are women held to higher standards? Evidence from peer review. https://www.erinhengel.com/research/publishing_female.pdf