⌘ S

Analysis of PennSound speech-to-text

includes additional analysis

```julia
using DataFrames, CSV, Statistics, Dates, Plots, Distributions,RollingFunctions
```

Sample Characteristics

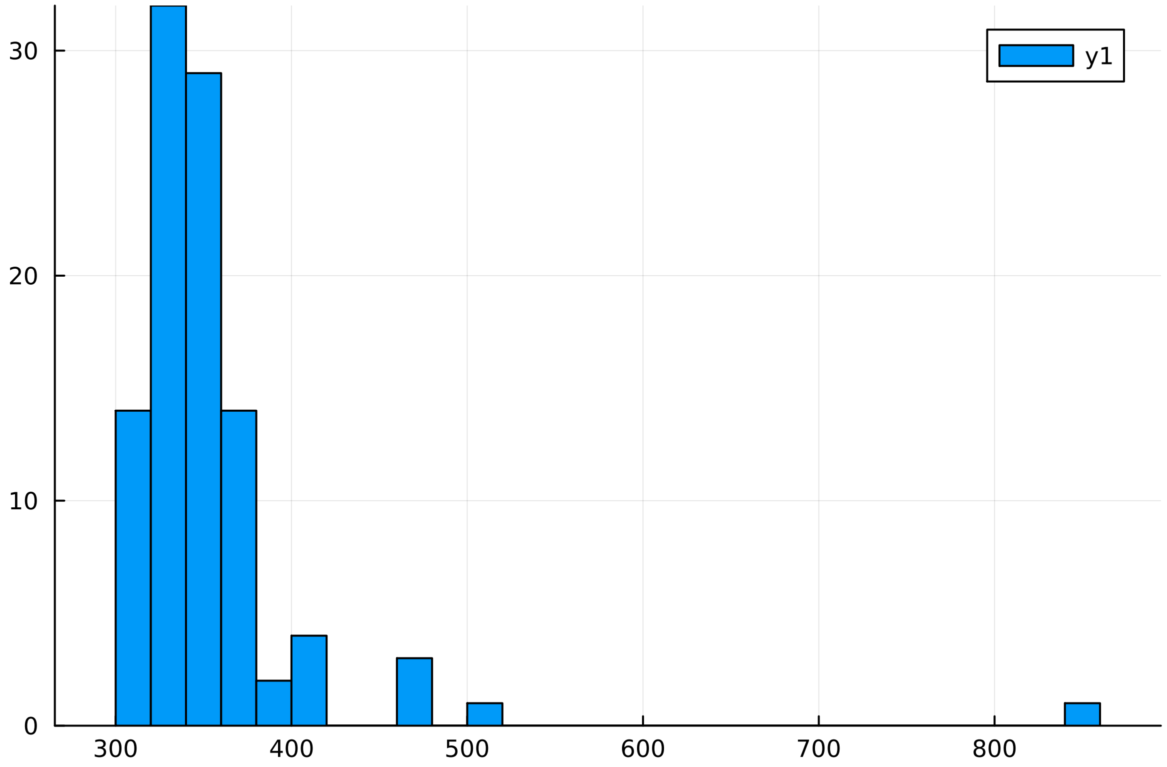
The original goal was a random sample of 100 clips, each 5 minutes in duration, which would be 8.3 hours of audio if done exactly that way. The total audio duration is 12.15 hours, which would mean quite a lot of silence if the speech was only 8.3 hours. However, the reference transcripts tend to indicate more than 5 minutes of speech, due to speech undetected by SAD, overlapping speech, or possibly human error (segments padded with silence). Summing the segment lengths from the human transcripts gives a total of 9.84 hours of speech, with a mean of 354 seconds and a standard deviation of 61.5 seconds. The histogram below shows the amount of speech per file in seconds.

speech =

	file	speech
1	"Andrews-Bruce-and-Charles-North_Compl	341.623
2	"Antin-David_Complete_Seminar_Universi	372.994
3	"Ashbery-John_01_Complete-Reading_WBAI	343.004
4	"Ashbery-John_Complete-Reading_Contemp	348.673
5	"Ashbery-John_Complete-Recording_Attit	300.015
6	"Ashbery-John_Complete-Recording_Honor	361.438
7	"Ashbery-John_Complete-Recording_Pione	315.977
8	"Ashbery-John_Complete-Recording_St-Ma	354.344
9	"Ashbery-John_Complete-Recording_The-S	336.321
10	"Ashbery-John_Complete-Recording_WBAI-	363.231
more		
100	"Yau-John_02_Complete-Reading_SUNY-Buf	337.962

```julia
speech = CSV.read("speech.tsv", DataFrame, delim="\t")

```



```
histogram(speech[:,:speech])
```

```
354.16752999999994
```

```
mean(speech.speech)
```

```
61.48031766484953
```

```
std(speech.speech)
```

	file	speech
1	"Ginsberg-Allen_Complete-Reading_WCW-L	858.552

```
speech[speech.speech .> 800,:]
```

```
9.837986944444443
```

```
sum(speech.speech) / 3600
```

total duration of audio clips is 12.15 hours durations =

	file	duration
1	"Andrews-Bruce-and-Charles-North_Compl	585.15
2	"Antin-David_Complete_Seminar_Universi	462.47
3	"Ashbery-John_01_Complete-Reading_WBAI	432.82
4	"Ashbery-John_Complete-Reading_Contemp	446.79
5	"Ashbery-John_Complete-Recording_Attit	390.93
6	"Ashbery-John_Complete-Recording_Honor	458.35
7	"Ashbery-John_Complete-Recording_Pione	393.12
8	"Ashbery-John_Complete-Recording_St-Ma	466.21
9	"Ashbery-John_Complete-Recording_The-S	440.09
10	"Ashbery-John_Complete-Recording_WBAI-	458.39
more		
100	"Yau-John_02_Complete-Reading_SUNY-Buf	379.11

```
durations = CSV.read("durations.tsv", DataFrame, delim="\t")

```

```
12.152402777777779
```

```
sum(durations.duration) / 3600
```

Word Error Rates

here we display individual WERs in various ways

	vs	azure	google	ibm	nemo	rev	whisper	whispercpp	nsp	snr
1	0	23.1	24.7	33.4	18.8	19.0	19.1	21.6	1	33.63
2	7	18.1	16.4	21.2	20.3	15.0	19.4	21.2	8	6.94
3		4.3	4.2	5.8	3.4	3.7	2.8	3.5	1	20.8
4		4.6	4.2	6.6	4.4	3.2	3.8	4.6	1	20.02
5		4.8	5.5	6.2	4.9	3.2	4.7	5.3	2	7.43
6	5	14.9	15.6	17.4	18.2	15.6	18.1	17.2	2	21.63
7		6.3	4.8	9.1	5.7	3.2	3.9	4.7	1	14.83
8		2.4	4.1	3.6	2.8	2.4	2.0	6.4	1	19.02
9		4.8	5.2	11.4	3.5	4.2	3.0	3.6	1	-0.31
10		4.5	4.6	8.6	4.7	4.2	3.5	5.8	1	12.39
more										
100		6.0	7.4	10.7	4.4	4.5	3.6	4.9	1	18.16

```julia
wers = CSV.read("wer.tsv", DataFrame, delim="\t")

```

substitution error rates only

	file	aws	azure	google	ibm	nemo	rev	whisper	whispercpp
66	erbach1"	5.1	4.2	4.5	7.2	3.1	4.8	2.6	3.5
67	erbach2"	3.1	3.4	4.3	5.7	2.7	4.6	3.0	2.9
68	'	4.7	4.7	5.4	7.7	4.5	4.8	3.9	4.7
69	ove"	4.4	4.0	5.1	5.7	3.4	3.6	3.3	3.8
70	"	2.9	3.7	4.0	6.9	2.9	3.2	2.6	2.9
71	ey"	9.1	11.8	9.8	18.4	7.4	8.4	9.1	8.6
72	"	11.5	13.1	12.1	17.7	10.4	10.6	10.3	10.4
73	ytalks1"	5.1	4.4	4.6	7.6	3.6	5.1	4.3	4.8
74	ytalks10"	4.7	5.9	4.2	6.7	3.0	2.9	3.5	3.3
75	ytalks3"	4.5	4.5	4.7	6.6	3.1	4.0	2.0	2.7
76	ytalks5"	2.7	2.0	1.9	3.1	1.5	2.6	1.6	1.9
77	ino"	2.7	2.0	2.8	4.5	1.2	3.3	2.4	2.3
78	alk"	3.7	4.1	4.5	6.5	3.8	4.1	3.2	2.8
79	th"	9.4	9.3	11.3	15.2	11.3	10.4	9.1	9.0
80	ack"	10.9	9.6	9.3	15.0	8.5	6.7	7.0	7.9
81	koff"	4.5	3.4	4.7	7.8	2.5	3.2	2.5	2.0
82	rds"	3.1	3.1	2.2	7.4	0.6	1.9	1.5	2.3
83	son1"	1.7	1.8	2.7	3.3	1.1	1.3	0.9	1.7

```
wers_s = CSV.read("wer_s.tsv", DataFrame, delim="\t")

```

deletion error rates only

	file	aws	azure	google	ibm	nemo	rev	whisper	whis
64	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
65	"kyger"	10.6	11.3	10.1	12.8	9.3	5.6	7.3	6.7
66	"lauterbach1"	1.2	2.2	3.2	2.4	3.9	2.1	2.2	3.3
67	"lauterbach2"	1.4	2.1	1.6	2.1	5.2	1.2	3.7	7.0
68	"levy"	0.7	0.8	1.8	1.2	2.0	0.9	2.2	1.3
69	"mirakove"	1.0	0.5	1.2	1.3	2.0	0.9	1.1	1.3
70	"moore"	0.0	0.1	1.4	0.5	0.5	0.9	0.0	0.6
71	"moxley"	5.0	3.6	4.5	3.5	4.1	3.8	1.9	3.1
72	"oppen"	3.2	3.8	4.8	5.7	3.7	3.9	1.7	4.0
73	"phillytalks1"	7.8	9.5	9.5	13.3	13.6	6.0	8.6	10.7
74	"phillytalks10"	0.9	1.3	1.1	1.8	1.4	1.0	1.5	1.1
75	"phillytalks3"	1.1	2.7	2.0	2.2	6.4	1.9	5.7	5.9
76	"phillytalks5"	8.1	11.4	8.7	10.4	13.3	5.1	10.2	7.4
77	"piombino"	1.1	0.6	1.5	0.9	1.7	0.6	1.4	2.0
78	"poemtalk"	5.7	4.9	3.9	5.1	9.5	3.2	5.5	5.5
79	"raworth"	0.8	1.3	4.0	2.3	1.4	1.1	1.1	1.6
80	"retalack"	8.0	10.3	10.8	10.9	15.0	10.4	13.9	14.0
81	"reznikoff"	1.2	2.4	2.5	3.1	3.1	3.0	2.1	2.5

```
wers_d = CSV.read("wer_d.tsv", DataFrame, delim="\t")

```

insertion error rates only

	file	aws	azure	google	ibm	nemo	rev	whisper	whis
83	"robinson1"	0.6	0.6	0.4	0.3	0.4	0.1	0.7	0.2
84	"robinson2"	1.0	0.9	1.0	0.7	0.8	0.5	0.6	0.8
85	"robinson3"	1.2	0.9	0.7	0.8	1.2	0.4	0.5	0.7
86	"rothenberg"	1.1	0.5	1.2	1.2	1.1	0.9	1.1	7.5
87	"scalapino1"	0.8	0.7	0.5	0.9	0.4	0.7	0.5	0.7
88	"scalapino2"	1.4	0.9	1.4	1.2	1.3	0.9	1.8	3.2
89	"sherlock"	1.1	1.4	0.3	1.2	1.0	0.6	0.9	0.8
90	"silliman1"	0.5	0.6	0.7	0.8	0.3	1.0	2.1	1.3
91	"silliman2"	0.8	0.8	0.8	0.8	1.1	1.3	1.4	0.9
92	"silliman3"	0.7	0.7	0.5	1.0	0.5	0.2	0.7	0.8
93	"smith"	2.0	1.2	1.6	1.2	1.2	1.2	0.6	0.6
94	"spahr"	1.5	1.4	1.1	1.2	0.8	0.8	0.9	0.8
95	"sze"	1.3	0.8	1.0	0.8	0.7	0.7	0.8	0.4
96	"templeton"	1.5	1.6	1.0	0.8	2.4	3.5	3.3	3.9
97	"torres"	6.4	5.0	5.3	5.6	3.9	6.7	3.7	12.2
98	"towle"	0.6	0.5	0.7	1.2	0.1	0.2	0.4	0.1
99	"wisher"	0.6	1.2	0.8	1.2	0.8	0.3	0.4	0.3

```julia
wers_i = CSV.read("wer_i.tsv", DataFrame, delim="\t")

```

sort by rev, the best performer

```
wers_sorted_by_rev =
```

	file	aws	azure	google	ibm	nemo	rev	whisper	whisperc
1	"bromige2"	2.4	3.0	2.0	4.2	2.4	1.1	2.0	1.9
2	"duplessis1"	3.0	2.9	2.9	4.4	3.9	1.8	1.9	5.4
3	"duplessis2"	2.8	2.6	3.6	6.7	2.9	2.0	1.8	1.6
4	"gladman"	2.2	2.4	3.2	4.1	4.2	2.2	1.9	10.9
5	"jarnot"	4.3	4.0	4.5	9.3	5.5	2.3	3.1	4.3
6	"ashbery6"	2.0	2.4	4.1	3.6	2.8	2.4	2.0	6.4
7	"robinson1"	3.0	2.8	5.0	4.2	4.1	2.5	2.6	3.6
8	"richards"	3.9	3.9	2.8	8.2	2.8	2.8	3.4	7.1
9	"robinson3"	3.6	3.2	3.9	5.2	4.2	2.9	2.9	5.3
10	"ashbery2"	3.6	4.6	4.2	6.6	4.4	3.2	3.8	4.6
more									
100	"ginsberg"	37.2	36.3	36.2	44.0	34.4	33.3	32.4	39.4

```julia
wers_sorted_by_rev = sort(wers, [:rev])
```

add mean WER to table

```
wers_with_mean =
```

	file	aws	azure	google	ibm	nemo	rev	whisper	whispercpp
1	"andrews"	29.0	23.1	24.7	33.4	18.8	19.0	19.1	21.6
2	"antin"	15.7	18.1	16.4	21.2	20.3	15.0	19.4	21.2
3	"ashbery1"	3.5	4.3	4.2	5.8	3.4	3.7	2.8	3.5
4	"ashbery2"	3.6	4.6	4.2	6.6	4.4	3.2	3.8	4.6
5	"ashbery3"	3.9	4.8	5.5	6.2	4.9	3.2	4.7	5.3
6	"ashbery4"	14.6	14.9	15.6	17.4	18.2	15.6	18.1	17.2
7	"ashbery5"	5.4	6.3	4.8	9.1	5.7	3.2	3.9	4.7
8	"ashbery6"	2.0	2.4	4.1	3.6	2.8	2.4	2.0	6.4
9	"ashbery7"	4.1	4.8	5.2	11.4	3.5	4.2	3.0	3.6
10	"ashbery8"	4.5	4.5	4.6	8.6	4.7	4.2	3.5	5.8
	more								
100	"yau"	5.5	6.0	7.4	10.7	4.4	4.5	3.6	4.9

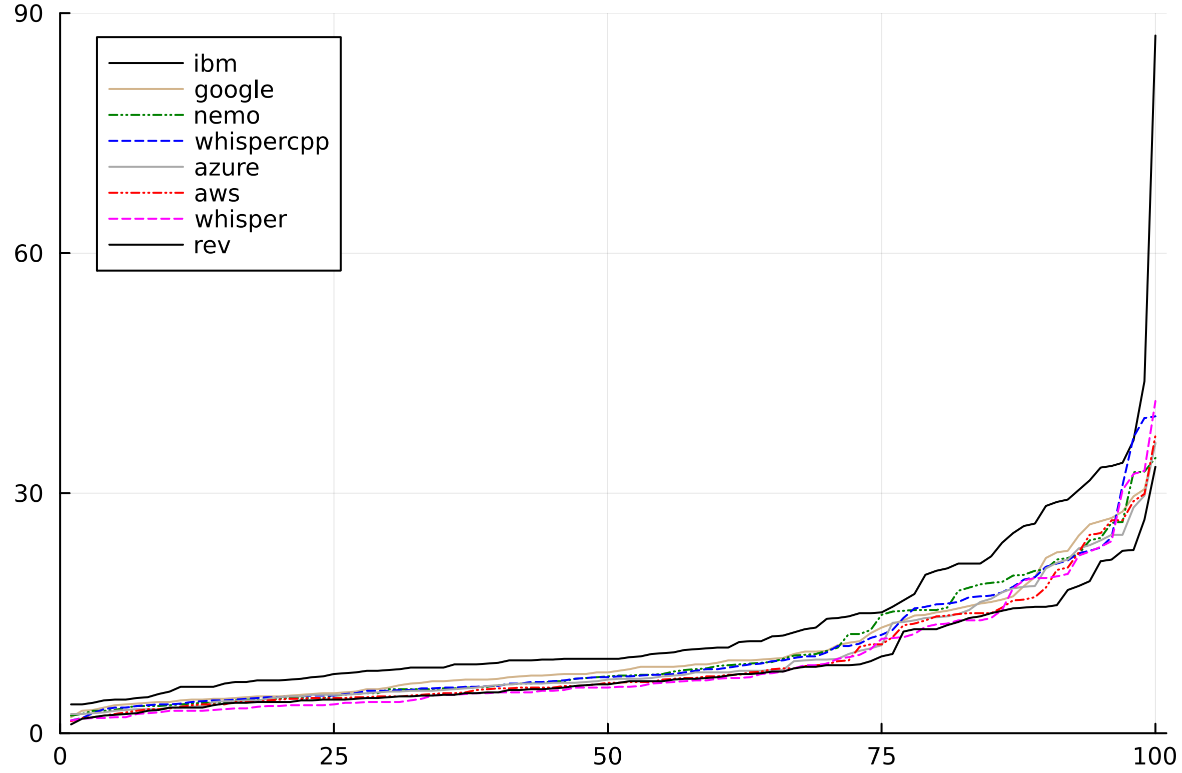
```
wers_with_mean = transform(wers, AsTable([:aws, :azure, :google, :ibm, :nemo, :rev,
:whisper, :whispercpp]) => ByRow(mean) => :mean)
```

sort by mean WER

	file	aws	azure	google	ibm	nemo	rev	whisper	whis
25	"ashbery7"	4.1	4.8	5.2	11.4	3.5	4.2	3.0	3.6
26	"fiedler"	5.0	5.3	4.3	8.2	6.2	3.8	3.4	4.0
27	"ashbery8"	4.5	4.5	4.6	8.6	4.7	4.2	3.5	5.8
28	"howe1"	5.9	4.8	4.3	10.1	4.8	3.9	3.9	3.4
29	"coolidge"	4.2	5.4	6.6	6.6	5.6	5.1	3.5	5.3
30	"ashbery5"	5.4	6.3	4.8	9.1	5.7	3.2	3.9	4.7
31	"drucker2"	4.6	5.3	7.0	7.0	3.6	6.1	3.8	5.7
32	"hawkins"	3.3	3.0	4.9	3.6	3.8	4.1	3.1	17.6
33	"wisher"	4.7	5.3	6.0	7.8	6.2	5.4	3.9	4.5
34	"bellamy"	4.5	4.6	8.6	5.8	4.4	5.5	5.0	6.4
35	"sze"	6.9	5.1	6.8	8.7	5.5	5.5	3.9	4.4
36	"yau"	5.5	6.0	7.4	10.7	4.4	4.5	3.6	4.9
37	"mirakove"	5.7	5.2	7.4	7.6	6.3	5.1	4.8	5.4
38	"davies"	5.6	5.0	5.0	8.8	5.5	4.7	4.9	8.7
39	"silliman1"	4.6	5.9	5.1	9.3	7.0	4.6	6.5	6.2
40	"garrison"	6.0	6.2	6.5	7.9	5.5	6.8	4.8	5.9
41	"bromige4"	4.9	6.9	6.5	9.2	5.8	5.0	5.7	5.6

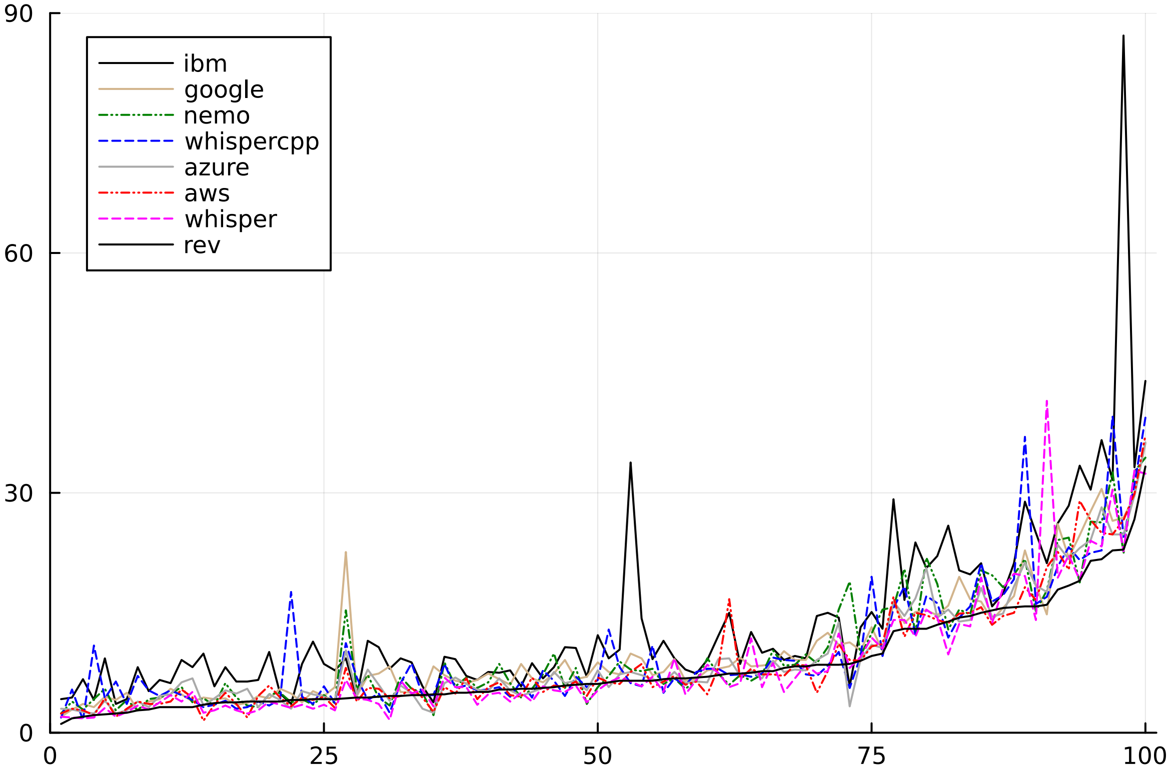
```
wers_sorted_by_mean = sort(wers_with_mean, :mean)
```

WERs sorted separately



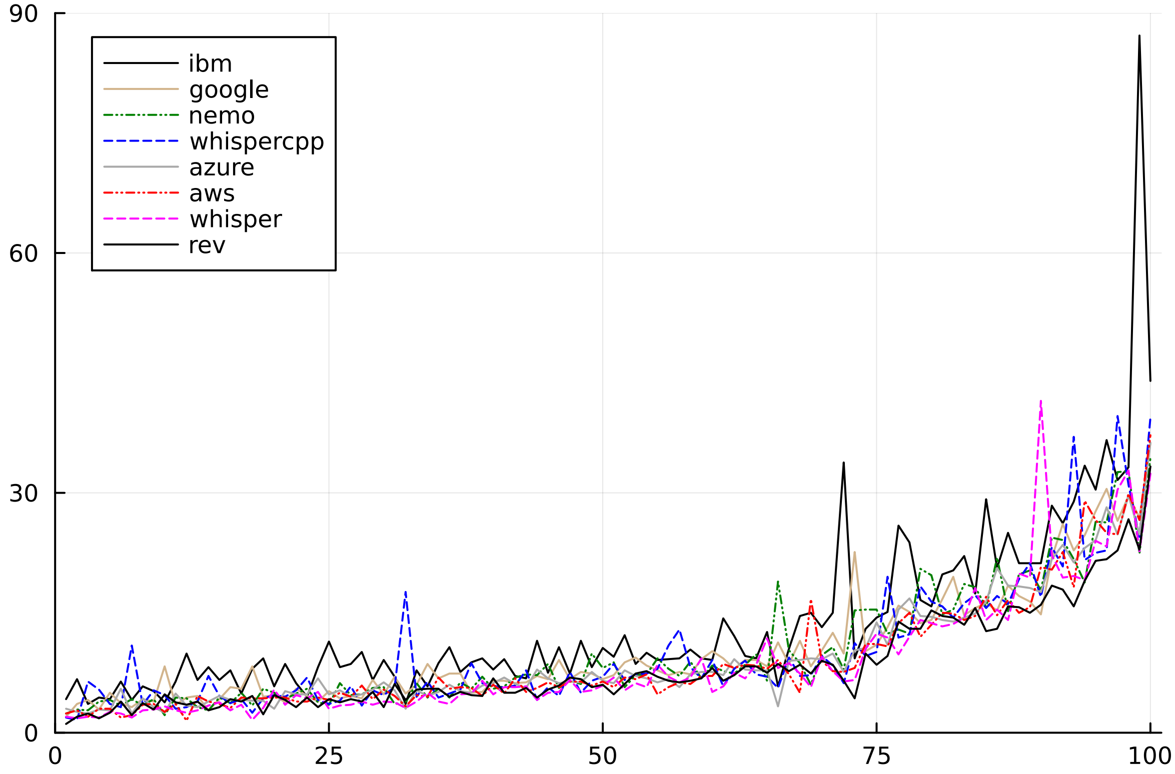
```julia
begin
    plot(sort(wers.ibm),label="ibm", linestyle=:solid, color=:black)
    plot!(sort(wers.google),label="google", linestyle=:solid, color=:tan)
    plot!(sort(wers.nemo),label="nemo", linestyle=:dashdotdot, color=:green)
    plot!(sort(wers.whispercpp),label="whispercpp", linestyle=:dash, color=:blue)
    plot!(sort(wers.azure),label="azure", linestyle=:solid, color=:darkgray)
    plot!(sort(wers.aws),label="aws",xlim=(0,101),ylim=(0,90),linestyle=:dashdotdot,color=:red)
    plot!(sort(wers.whisper),label="whisper", linestyle=:dash, color=:magenta)
    plot!(sort(wers.rev),label="rev", linestyle=:solid,color=:black)
end
```

WERs sorted by rev (best system)



```julia
begin
    w1 = wers_sorted_by_rev
    plot(w1.ibm,label="ibm", linestyle=:solid, color=:black)
    plot!(w1.google,label="google", linestyle=:solid, color=:tan)
    plot!(w1.nemo,label="nemo", linestyle=:dashdotdot, color=:green)
    plot!(w1.whispercpp,label="whispercpp", linestyle=:dash, color=:blue)
    plot!(w1.azure,label="azure", linestyle=:solid, color=:darkgray)
    plot!(w1.aws,label="aws",xlim=(0,101),ylim=(0,90), linestyle=:dashdotdot,color=:red)
    plot!(w1.whisper,label="whisper", linestyle=:dash, color=:magenta)
    plot!(w1.rev,label="rev", linestyle=:solid,color=:black)
end
```

WERs sorted by mean WER



```julia
begin
    w2 = wers_sorted_by_mean
    plot(w2.ibm,label="ibm", linestyle=:solid, color=:black)
    plot!(w2.google,label="google", linestyle=:solid, color=:tan)
    plot!(w2.nemo,label="nemo", linestyle=:dashdotdot, color=:green)
    plot!(w2.whispercpp,label="whispercpp", linestyle=:dash, color=:blue)
    plot!(w2.azure,label="azure", linestyle=:solid, color=:darkgray)
    plot!(w2.aws,label="aws",xlim=(0,101),ylim=(0,90), linestyle=:dashdotdot,color=:red)
    plot!(w2.whisper,label="whisper", linestyle=:dash, color=:magenta)
    plot!(w2.rev,label="rev", linestyle=:solid,color=:black)
end
```

Diarization Error Rates

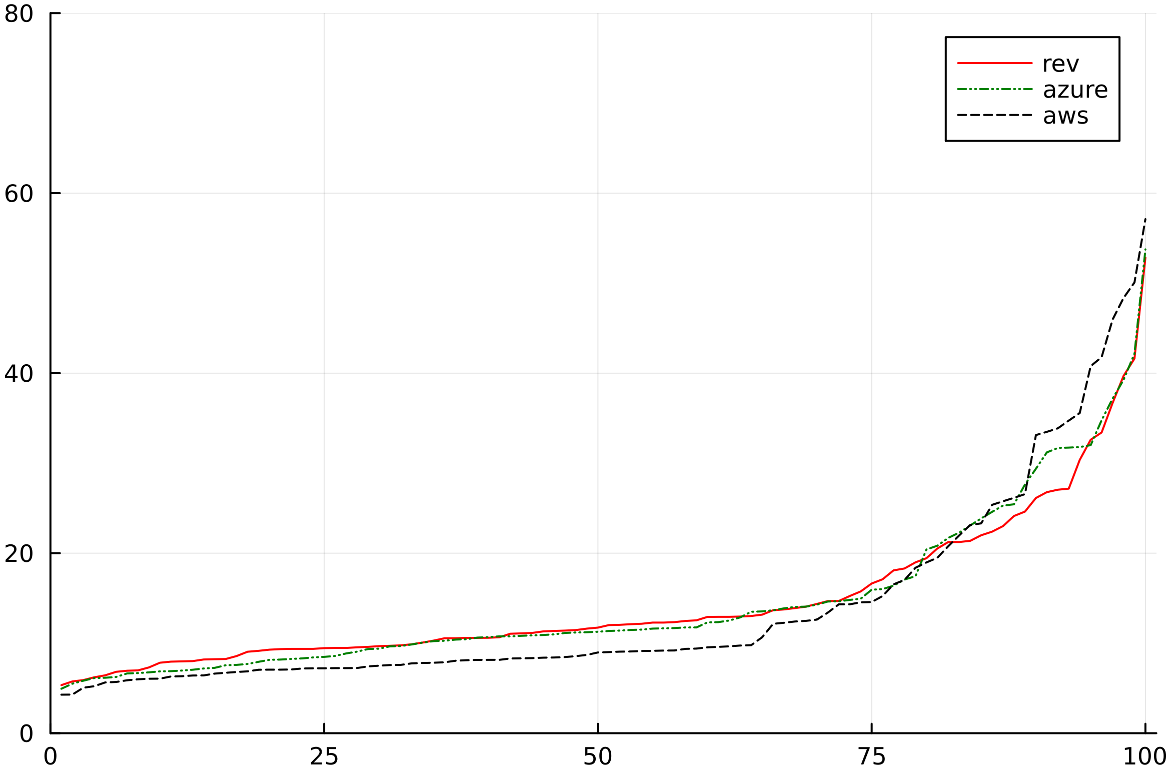
der =

	file	aws	azure	ibm	rev
1	"Andrews-Bruce-and-Charles-North_Compl	16.54	13.53	62.37	23.01
2	"Antin-David_Complete_Seminar_Universi	20.78	25.43	65.45	27.17
3	"Ashbery-John_01_Complete-Reading_WBAI	9.59	10.62	15.08	10.59
4	"Ashbery-John_Complete-Reading_Contemp	8.15	11.36	47.78	12.47
5	"Ashbery-John_Complete-Recording_Attit	8.05	10.87	18.4	11.73
6	"Ashbery-John_Complete-Recording_Honor	21.98	31.79	67.19	24.62
7	"Ashbery-John_Complete-Recording_Pione	8.41	8.18	20.46	6.82
8	"Ashbery-John_Complete-Recording_St-Ma	9.08	11.47	53.98	9.37
9	"Ashbery-John_Complete-Recording_The-S	8.15	8.48	59.48	9.37
10	"Ashbery-John_Complete-Recording_WBAI-	8.97	11.27	23.86	12.54
more					
100	"Yau-John_02_Complete-Reading_SUNY-Buf	7.06	7.69	8.13	9.47

```julia
der = CSV.read("der.tsv", DataFrame, delim="\t")

```

DERs sorted separately



```
begin
    plot(sort(der.rev),label="rev",linestyle=:solid,color=:red,xlim=(0,101),ylim=
    (0,80))
    plot!(sort(der.azure),label="azure", linestyle=:dashdotdot, color=:green)
    plot!(sort(der.aws),label="aws", linestyle=:dash, color=:black)
end
```

What about the outliers?

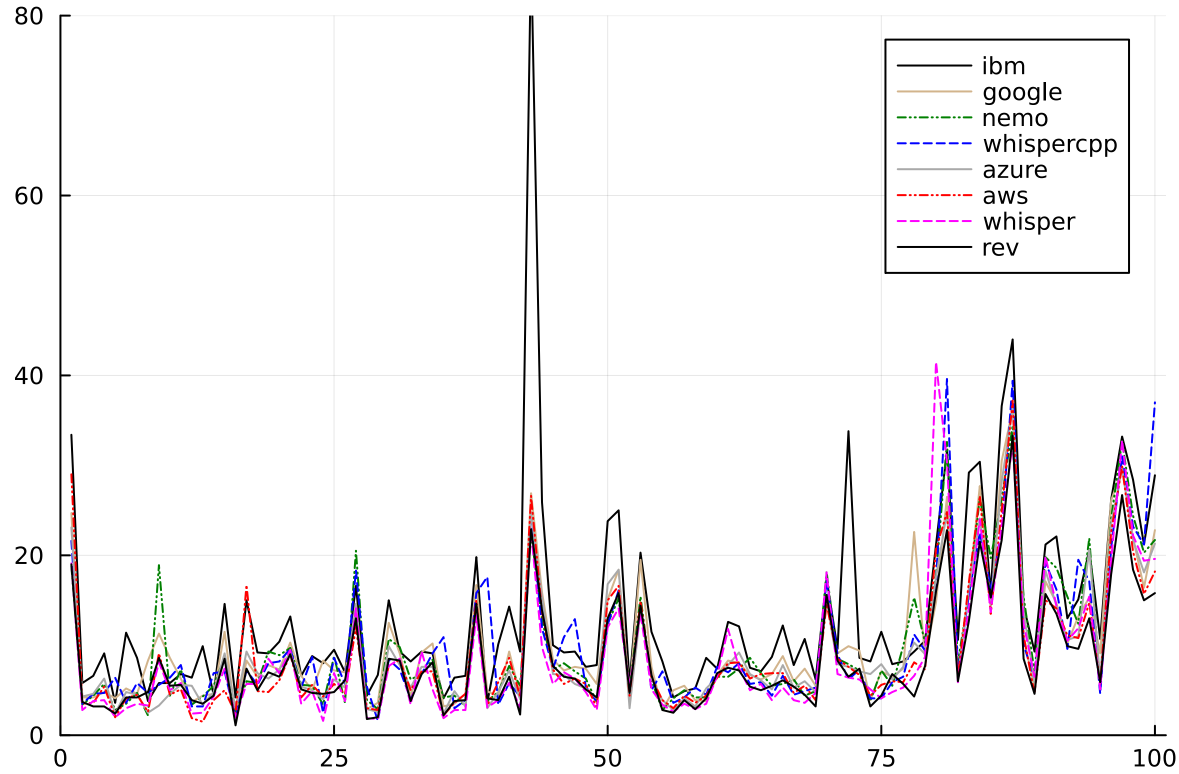
WERs sorted by SNR provided by IBM

```
wers_sorted_by_snr =
```

	file	aws	azure	google	ibm	nemo	rev	whisper	whispercpp
1	"andrews"	29.0	23.1	24.7	33.4	18.8	19.0	19.1	21.6
2	"ashbery1"	3.5	4.3	4.2	5.8	3.4	3.7	2.8	3.5
3	"ashbery2"	3.6	4.6	4.2	6.6	4.4	3.2	3.8	4.6
4	"ashbery5"	5.4	6.3	4.8	9.1	5.7	3.2	3.9	4.7
5	"ashbery6"	2.0	2.4	4.1	3.6	2.8	2.4	2.0	6.4
6	"ashbery7"	4.1	4.8	5.2	11.4	3.5	4.2	3.0	3.6
7	"ashbery8"	4.5	4.5	4.6	8.6	4.7	4.2	3.5	5.8
8	"auster"	2.6	2.5	8.3	3.8	2.2	4.8	3.3	4.7
9	"beaulieu"	9.1	3.3	11.3	5.8	18.9	8.6	8.1	5.6
10	"bellamy"	4.5	4.6	8.6	5.8	4.4	5.5	5.0	6.4
more									
100	"benson2"	18.2	21.3	22.8	28.9	21.7	15.8	19.6	37.0

```
wers_sorted_by_snr = sort(wers, [:nsp])
```

SNR seems to have an overall effect, but not consistently, and doesn't seem to explain the outliers



```
begin
    plot(wers_sorted_by_snr.ibm,label="ibm", linestyle=:solid, color=:black)
    plot!(wers_sorted_by_snr.google,label="google", linestyle=:solid, color=:tan)
    plot!(wers_sorted_by_snr.nemo,label="nemo", linestyle=:dashdotdot, color=:green)
    plot!(wers_sorted_by_snr.whispercpp,label="whispercpp", linestyle=:dash, color=:blue)
    plot!(wers_sorted_by_snr.azure,label="azure", linestyle=:solid, color=:darkgray)
    plot!(wers_sorted_by_snr.aws,label="aws",xlim=(0,101),ylim=(0,80), linestyle=:dashdotdot,color=:red)
    plot!(wers_sorted_by_snr.whisper,label="whisper", linestyle=:dash, color=:magenta)
    plot!(wers_sorted_by_snr.rev,label="rev", linestyle=:solid,color=:black)
end
```

IBM specifically

what happened with ibm?

There's one extreme outlier, *joris*, where most words are missing. This recording has a lot of distortion/feedback.

	file	aws	azure	google	ibm	nemo	rev	whisper	whispercpp	ns
1	"joris"	26.6	24.8	26.9	87.2	22.4	22.9	22.7	24.5	1

```
wers[ wers.file .== "joris", :]
```

Mark, Neville, James, You can listen to it here

Another outlier is *corrigan*

	file	aws	azure	google	ibm	nemo	rev	whisper	whispercpp
1	"corrigan"	7.6	7.6	9.9	33.8	7.9	6.5	6.4	6.2

```
wers[wers.file .== "corrigan", :]
```

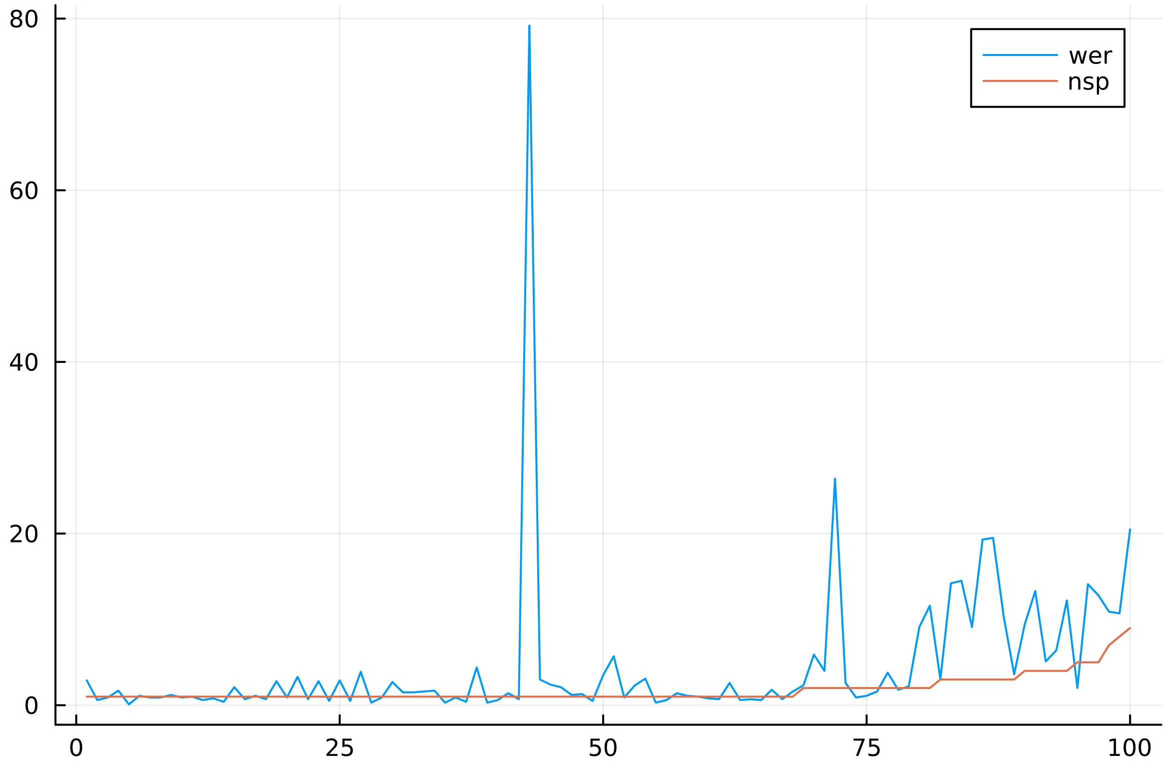
this file is unremarkable, but also has a lot of deletions. E.g., there's a 15s region from 100s to 115s that's missing, which is where a second speaker joins.

	file	aws	azure	google	ibm	nemo	rev	whisper	whispercpp
1	"corrigan"	1.6	1.4	3.0	26.4	3.3	1.3	1.3	1.4

```
wers_d[ wers_d.file .== "corrigan", :]
```

so IBM seems to have an issue with deletions, maybe due to speakers. sorting by number of speakers shows some effect. The first 68 have a single speaker. *corrigan* has two speakers, so this doesn't seem total explanatory for it's high DER.

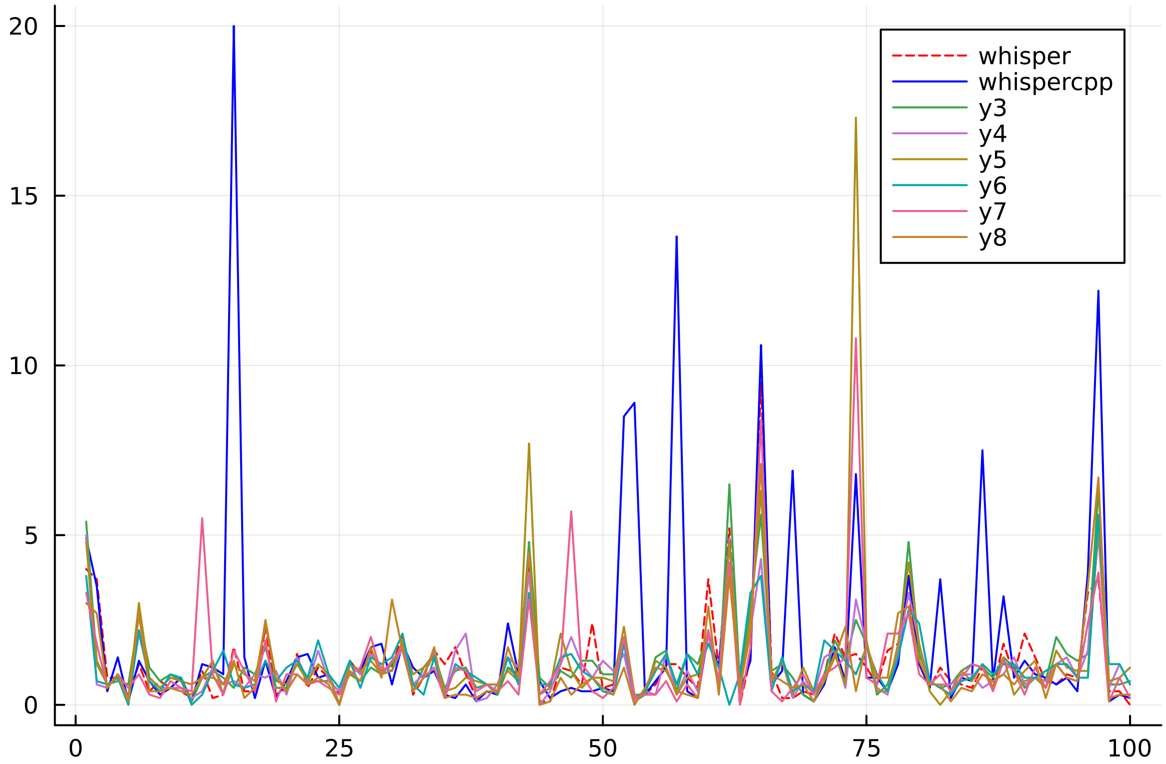
```
md"so IBM seems to have an issue with deletions, maybe due to speakers.  sorting by
number of speakers shows some effect.  The first 68 have a single speaker.
*corrigan* has two speakers, so this doesn't seem totally explanatory for it's high
DER."
```



```
begin
    wers_dn = sort(wers_d, [:nsp])
    plot(wers_dn.ibm, label="wer")
    # plot!(wers_d.ibm)
    plot!(wers_dn.nsp,label="nsp")
    # plot!(wers.snr,label="snr")
end
```

Whisper and Whisper cpp specifically

whispercpp (dark blue solid) stands out among IERs due to hallucinations, but *whisper* (red dashed) does not. *whispercpp* doesn't have the new options to limit hallucinations



```julia
begin
    wm = wers_i
    plot(wm.whisper, label="whisper", color=:red, linestyle=:dash)
    plot!(wm.whispercpp, label="whispercpp", color=:blue)
    plot!(wm.aws)
    plot!(wm.azure)
    plot!(wm.google)
    plot!(wm.ibm)
    plot!(wm.nemo)
    plot!(wm.rev)
end
```

Google specifically

```julia
md"### Google specifically"
```

phillytalk10 is an outlier for google with a very high IER. There's a long string of individual digits inserted during a period of silence (no speech), similar to a hallucination in whisper, although it's only digits.

	file	aws	azure	google	ibm	nemo	rev	whisper	whisperc
1	"phillytalks10"	2.5	3.1	17.3	0.9	10.8	0.4	1.5	6.8

```julia
wers_i[wers_i.file .== "phillytalks10", :]
```