{Awk als Excel-Ersatz — Cheat Sheet}

Jürgen Key
https://elbosso.github.io/excel__nein___awk_gnuplot_.html#content

Example setup

- Make test data (cut off first lines with ls garbage)

```
ls -l /usr/bin |tail -n +2
```

- Number of lines using wc

```
ls -l /usr/bin |tail -n +2| wc -l
```

- Number of lines using awk

```
ls -l /usr/bin |tail -n +2| awk 'END{print NR}'
```

- Numbered lines

```
ls -l /usr/bin |tail -n +2| awk '{print NR":"$5}'
```

Some statistics

- Average (mean) for column 5

```
ls -l /usr/bin |tail -n +2| awk 'BEGIN{sum=0.0}{sum+=$5}END{printf("Average: %f\n",sum/NR)}'
```

- Variables are initialized to 0 - so average of column N with $n = 5$

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{sum+=$N}END{printf("Average: %f\n",sum/NR)}'
```

- Standard deviation

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{sum+=$N;sumsq+=$N^2}END{printf("Average: %f Std deviation: %f\n",sum/NR,sqrt((sumsq-sum^2/NR)/NR))}'
```

- Median

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{print $N}'|sort -n| awk '{a[i++]=$1}' \
'END{x=int((i+1)/2);if (x < (i+1)/2) y=(a[x-1]+a[x])/2; else y=a[x-1]; printf("Median: %f\n",y)}'
```

- Percentiles

```
ls -l /usr/bin |tail -n +2| awk -v N=5 \
'{print $N}'|sort -n| awk '{s[NR-1]=$N} END{for(i=0.1;i<=1.0;i+=0.1){printf("%d %f\n",i*100,s[int(NR*i-0.5)])}}'
```

- Histogram

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{print $N}'|sort -n| awk -v DELTA=150000 \
'BEGIN{delta = (DELTA == ""? 10000 : DELTA)}' \
'{bucketNr = int(($0+delta) / delta);cnt[bucketNr]++;numBuckets = (numBuckets > bucketNr ? numBuckets : bucketNr)}' \
'END{for (bucketNr=1; bucketNr<=numBuckets; bucketNr++) {end = beg + delta;printf("%0.1f %0.1f %d\n"), beg, end, cnt[bucketNr];beg = end;}}'>hist.dat
```

Functions

- Function for rounding

```
func round(n)
{
  return int(n+0.5)
}
```

- Function for ceiling

```
func ceil(n)
{
  return n%1 ? int(n)+1 : n
}
```

{Awk als Excel-Ersatz — Cheat Sheet}

Jürgen Key
https://elbosso.github.io/excel__nein___awk_gnuplot_.html#content

Gnuplotting

- Only create histogram for the interesting part - note that there are now more than one variable on the command line

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{print $N}'|sort -n| awk -v DELTA=1000 -v MAX=MAX \
'BEGIN{delta = (DELTA == ""? 10000 : DELTA)} ' \
'{if($0<15001){bucketNr = int(($0+delta) / delta);cnt[bucketNr]++;numBuckets = (numBuckets > bucketNr ? numBuckets : bucketNr)}}' \
'END{for (bucketNr=1; bucketNr<=numBuckets; bucketNr++) {end = beg + delta;printf "%0.1f %0.1f %d\n", beg, end, cnt[bucketNr];beg = end;}}'>hist.dat
```

- Gnuplot

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{print $N}'|sort -n| awk -v DELTA=1000 -v MAX=MAX \
'BEGIN{delta = (DELTA == ""? 10000 : DELTA)} ' \
'{if($0<15001){bucketNr = int(($0+delta) / delta);cnt[bucketNr]++;numBuckets = (numBuckets > bucketNr ? numBuckets : bucketNr)}}' \
'END{for (bucketNr=1; bucketNr<=numBuckets; bucketNr++) {end = beg + delta;printf "%0.1f %0.1f %d\n", beg, end, cnt[bucketNr];beg = end;}}'| \
gnuplot -p -e "set terminal dumb size $(tput cols), $(tput lines) enhanced; set autoscale;set style data histogram;set style fill solid;"\
"plot '-' using 3:xtic(1)"
```

Conditionals

- Only use certain lines in computation using regular expressions as criterion

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{if($9 ~ /^m/){sum+=$N;sumsq+=$N^2;count++}}' \
'END{printf("Average: %f Std deviation: %f Count: %d\n",sum/count,sqrt((sumsq-sum^2/count)/count),count)}'
```

- Only use certain lines in computation using case insensitive regular expressions as criterion

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{IGNORECASE = 1;if($9 ~ /^m/){sum+=$N;sumsq+=$N^2;count++}}' \
'END{printf("Average: %f Std deviation: %f Count: %d\n",sum/count,sqrt((sumsq-sum^2/count)/count),count)}'
```

- Alternatively

```
ls -l /usr/bin |tail -n +2| awk -v N=5 'IGNORECASE = 1 && $9 ~ /^m/{sum+=$N;sumsq+=$N^2;count++}' \
'END{printf("Average: %f Std deviation: %f Count: %d\n",sum/count,sqrt((sumsq-sum^2/count)/count),count)}'
```

- Only use certain lines in computation depending on the line numbers

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '{if(NR >9 && NR<21){sum+=$N;sumsq+=$N^2;count++}}' \
'END{printf("Average: %f Std deviation: %f Count: %d\n",sum/count,sqrt((sumsq-sum^2/count)/count),count)}'
```

- Alternatively

```
ls -l /usr/bin |tail -n +2| awk -v N=5 '(NR >9 && NR<21){sum+=$N;sumsq+=$N^2;count++}' \
'END{printf("Average: %f Std deviation: %f Count: %d\n",sum/count,sqrt((sumsq-sum^2/count)/count),count)}'
```

- Only use certain lines in computation using timestamps as criterion

```
ls -lt --time-style=long-iso /usr/bin|awk -v date=2020-06-09 'date<$6{print $8}'
```

- Colorize cells depending on their value

```
ls -l /usr/bin |tail -n +2| \
awk '{if($5>999999)print NR"\t"$1"\t"$2"\t"$3"\t"$4"\t\033[1;31m"$5"\033[0m"; \
else print NR"\t"$1"\t"$2"\t"$3"\t"$4"\t"$5;}'
```