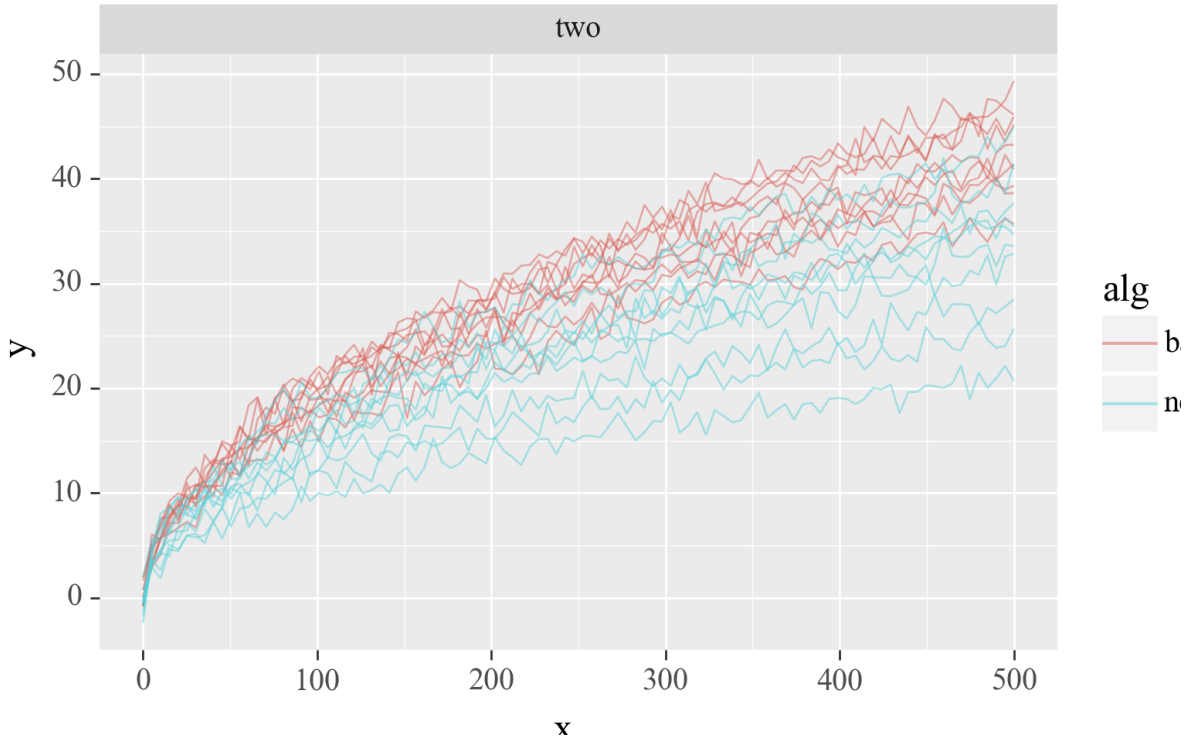
In [98]:
```
import numpy as np
import pandas as pd
import plotnine as gg

%matplotlib notebook
gg.theme_set(gg.theme_gray(base_size=18, base_family='times new roman'))
gg.theme_update(figure_size=(8,5))
```

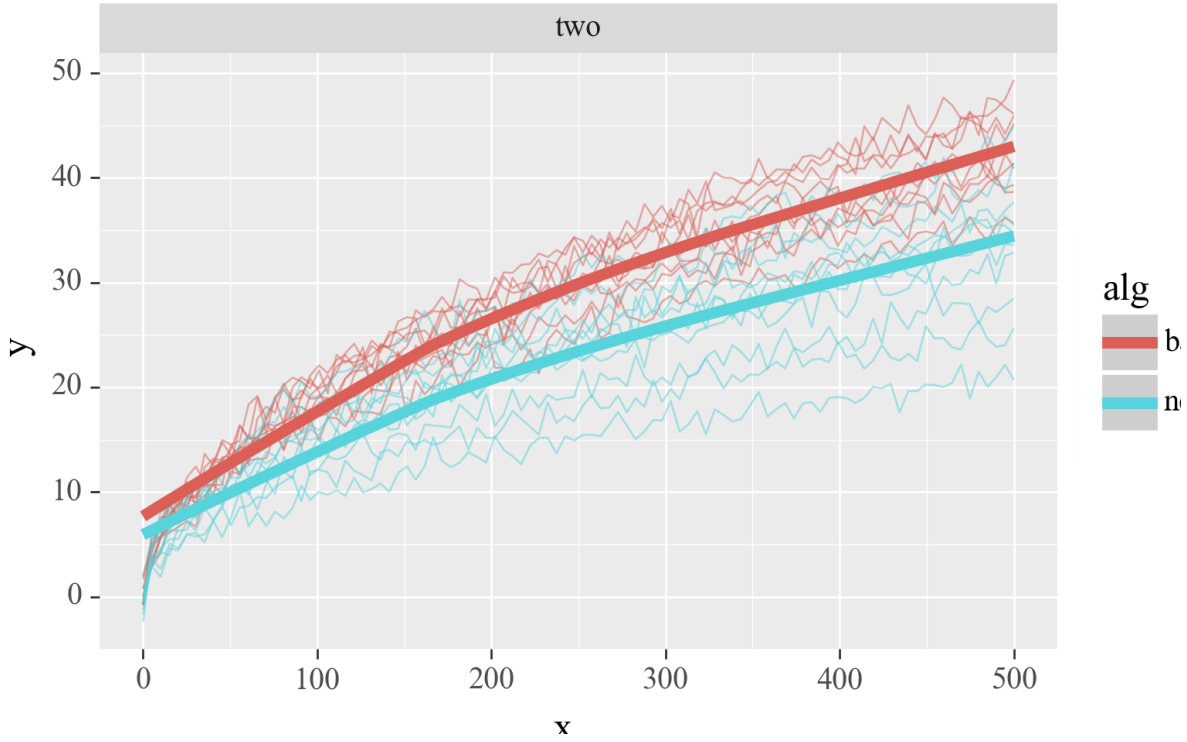
In [99]:
```
# Generating a fake dataset, inspired by some ML learning curves.
seeds = range(10)
algs = ['baseline', 'new']

x = np.linspace(0, 500, 100)

df_list = []
for alg in algs:
    scale *= (1 + 0.1 * np.random.randn())
    for seed in seeds:
        scale *= (1 + 0.1 * np.random.randn())
        df_list.append(pd.DataFrame({
                    'x': x,
                    'y': scale * x ** 0.5 + np.random.randn(len(x)),
                    'alg': alg,
                    'seed': seed}))

df = pd.concat(df_list)
df['unique_id'] = df.alg + df.seed.astype(str)
```

In [100]:
```
# Plotting everything works as expected
p = (gg.ggplot(df)
     + gg.aes('x', 'y', colour='alg')
     + gg.geom_line(gg.aes(group='unique_id'), alpha=0.5)
     + gg.facet_wrap('~ game', scales='free'))
p
```



Out[100]: <ggplot: (341316937)>

In [101]:
```
# If you want to add a smoothing everything also works fine
p + gg.geom_smooth(method='lowess', size=3)
```



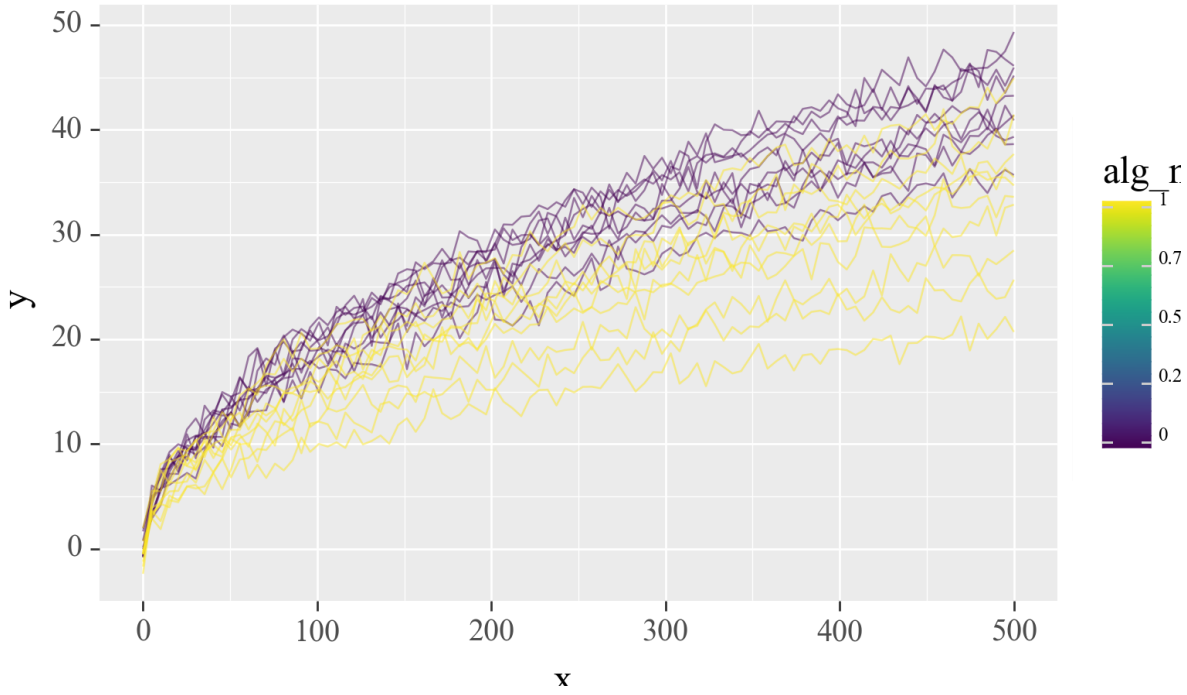
Out[101]: <ggplot: (335276653)>

In [102]:
```
# However, things mess up with numerical values instead of strings
df['alg_num'] = df.alg.astype('category').cat.codes.values
df.head()
```

Out[102]:

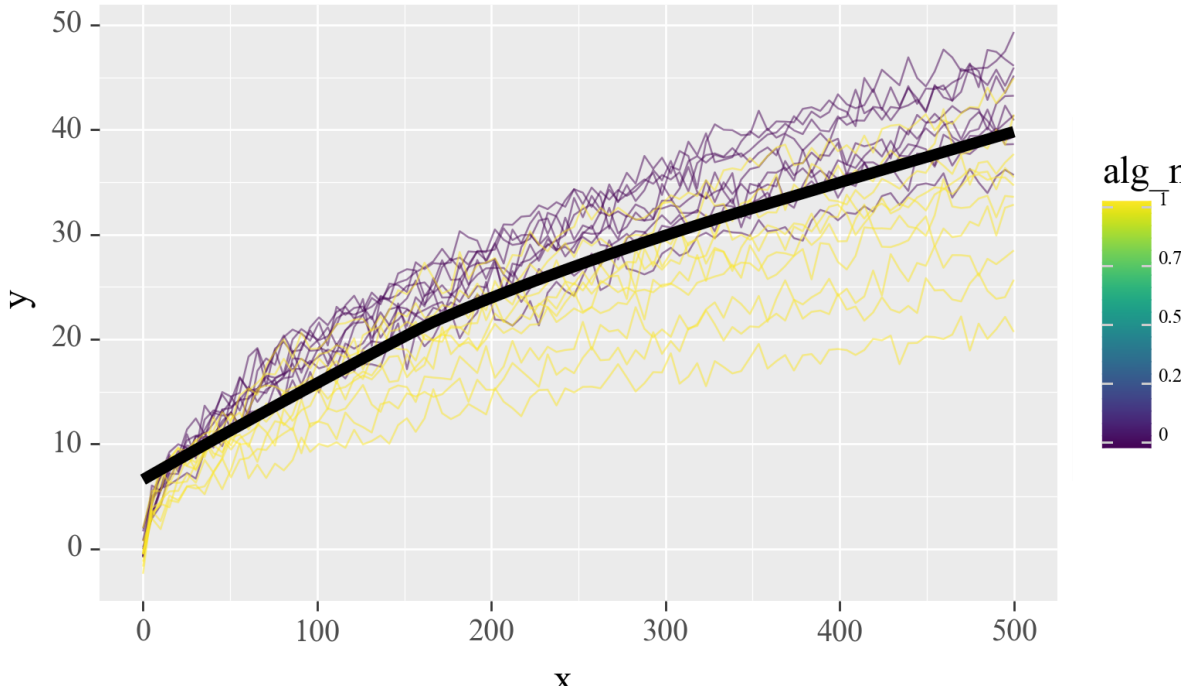
	alg	seed	x	y	unique_id	alg_num
0	baseline	0	0.000000	-0.840035	baseline0	0
1	baseline	0	5.050505	3.249571	baseline0	0
2	baseline	0	10.101010	5.746611	baseline0	0
3	baseline	0	15.151515	7.689841	baseline0	0
4	baseline	0	20.202020	9.336211	baseline0	0

In [103]:
```
# Grouping and colours are fine when numbers
p = (gg.ggplot(df)
     + gg.aes('x', 'y', colour='alg_num')
     + gg.geom_line(gg.aes(group='unique_id'), alpha=0.5))
p
```



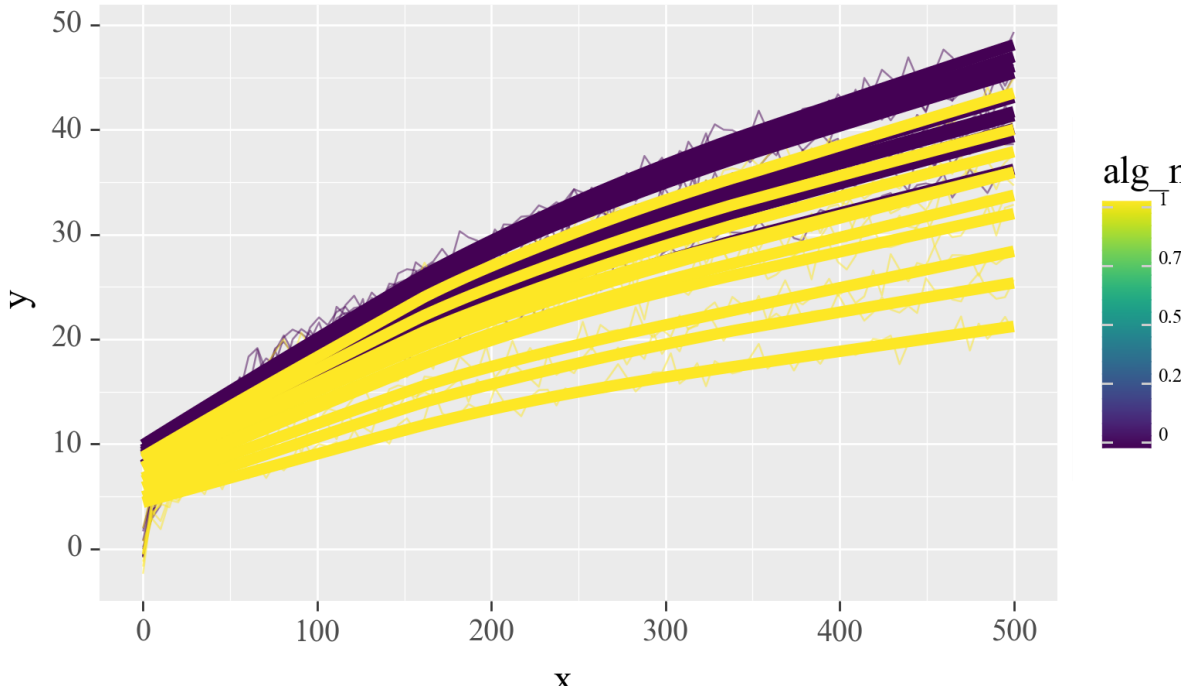
Out[103]: <ggplot: (338794681)>

In [104]:
```
# But smoothing is not automatically over the right groups!
# Also, colour is not properly propagated to the geom_smooth...
p + gg.geom_smooth(method='lowess', size=3)
```



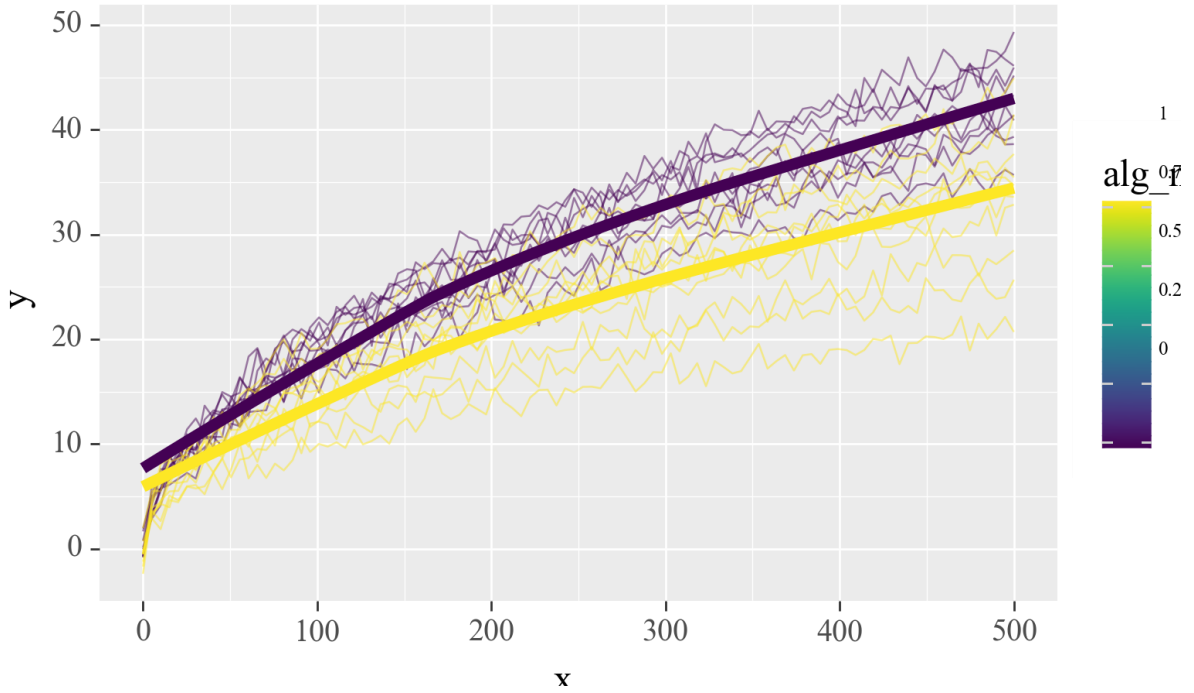
Out[104]: <ggplot: (338799345)>

In [105]:
```
# This problem can get even worse if you have arbitrary aes(...)
# Even if the aesthetic mappings have nothing to do with anything!
p + gg.geom_smooth(gg.aes(banana='unique_id'), size=3, method='lowess')
```



Out[105]: <ggplot: (345621261)>

In [106]:
```
# You can get around this by specifying the group...
# It would be nicer if this was automatic
p + gg.geom_smooth(gg.aes(banana='unique_id', group='alg_num'), size=3, metho
```



Out[106]: <ggplot: (345621445)>

In [ ]:
```
# I had a look at the code and the way "DISCRETE" variables are treated
# seems to be a little brittle... ggplot2 in R seemed to know a bit
# better about what to group over automatically.
```