RobustnessToChangesInModel

Rasmus Brostrøm

2024-04-10

The following data is generated by having the optimal threshold strategy and the data-driven strategy affect the underlying diffusion process and then observe the cumulative reward generated for each strategy, such that the regret can be calculated. The data has increasing time horizon going from 100 to 5000 with increments of 100. We consider drift functions of the form:

$$b(x) = -cx, \quad \text{with } c \in \{0.1, 0.5, 4\}$$

and reward functions of the form:

$$g(x) = a - |1-x|^p, \quad \text{with } p \in \{0.5, 1, 2, 5\}, \text{ and } a \in \{0.7, 0.9, 0.99\}$$

Each combination of drift function and reward function is simulated 100 times.

Different linear drift coefficients

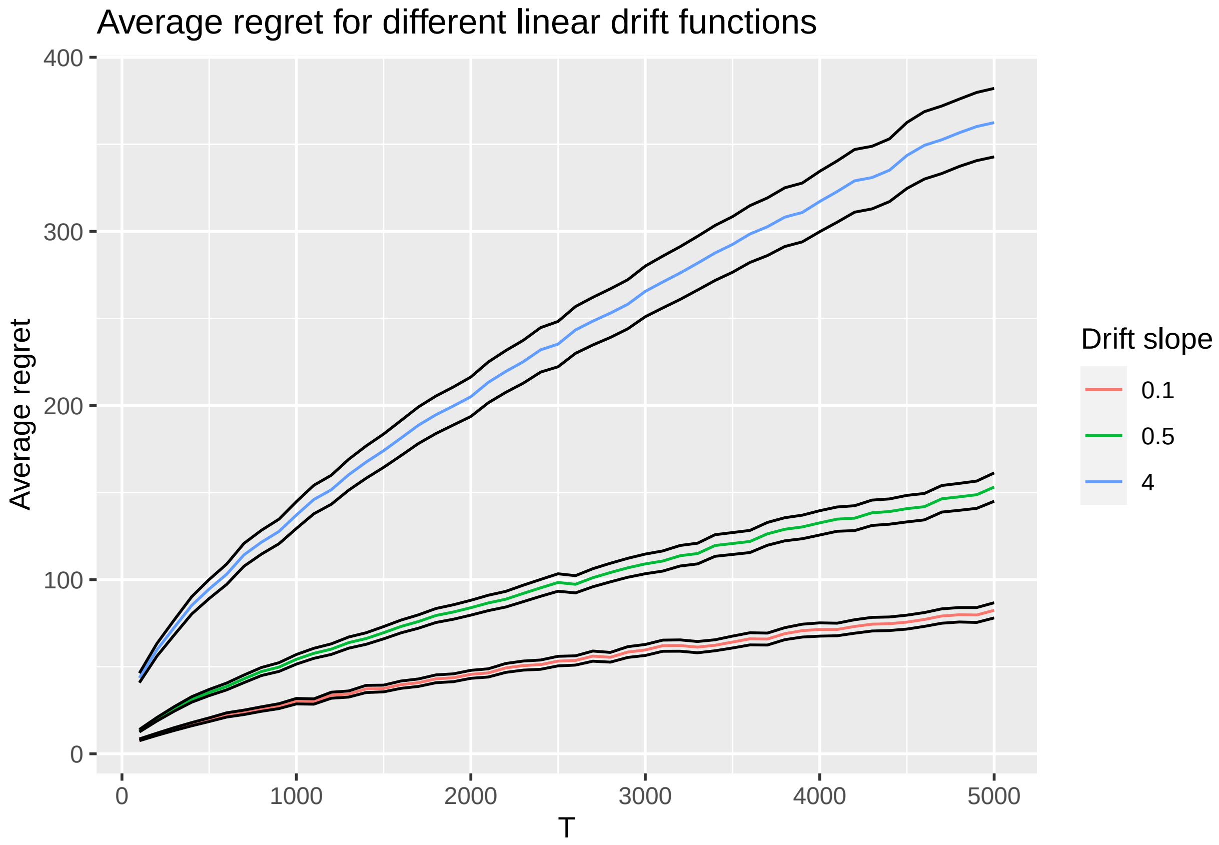
First, lets see how the different drift slope coefficients impacted the optimal threshold on average over all reward functions.

```
## # A tibble: 3 x 2
##       C avgOptThreshold
##   <dbl>           <dbl>
## 1   0.1           0.517
## 2   0.5           0.470
## 3   4             0.261
```

Here we see that on average the higher the linear coefficient is for the drift the lower the the optimal threshold is, which corresponds with how it affects the expected hitting times. As the linear coefficient increases, the expected hitting times goes from having linear growth to an exponential growth in the interval $[0, \zeta]$, meaning that on average it takes way too long for the process to get to certain values, and though the maximum reward might be achieved by stopping the process at a higher value it isn't beneficial, since the agent is able to stop the process several times and get the same cumulative reward with a lower threshold.

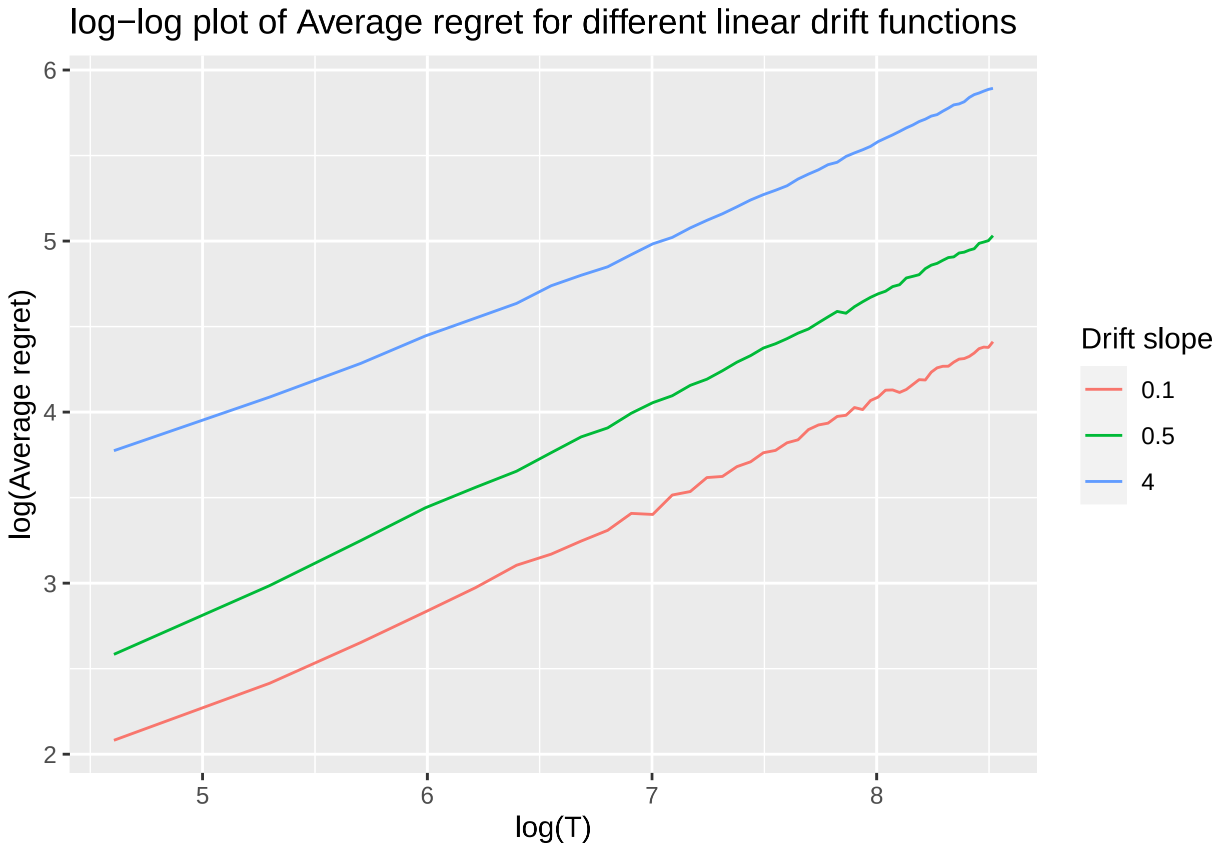
To see the general impact of the drift function on the data-driven algorithm I will look at the impact of the different drift coefficients over all of the reward functions first. Therefore, I group over all the different time horizons and the different coefficients for C.

Plotting the average regret lines with their 95 % confidence interval based on the standard error.



Here we see that the higher the linear drift, the higher the average regret is and the regret starts to take on the shape of the upper bound for the cumulative regret, while lower values of the slope coefficient for the linear drift makes the average regret have the form of the lower bound.

To get a better idea of how the exponents for the regret per time is changing, then we can use the log-log plot, which allows us to compare the exponents and the constant factor affecting the relationship between the average regret and the time horizon.



Here we see that increasing the slope of the drift does not really change the order of the regret, since all lines seem to have the same slope, but it does change the constant factor, since the lines have different interceptions.

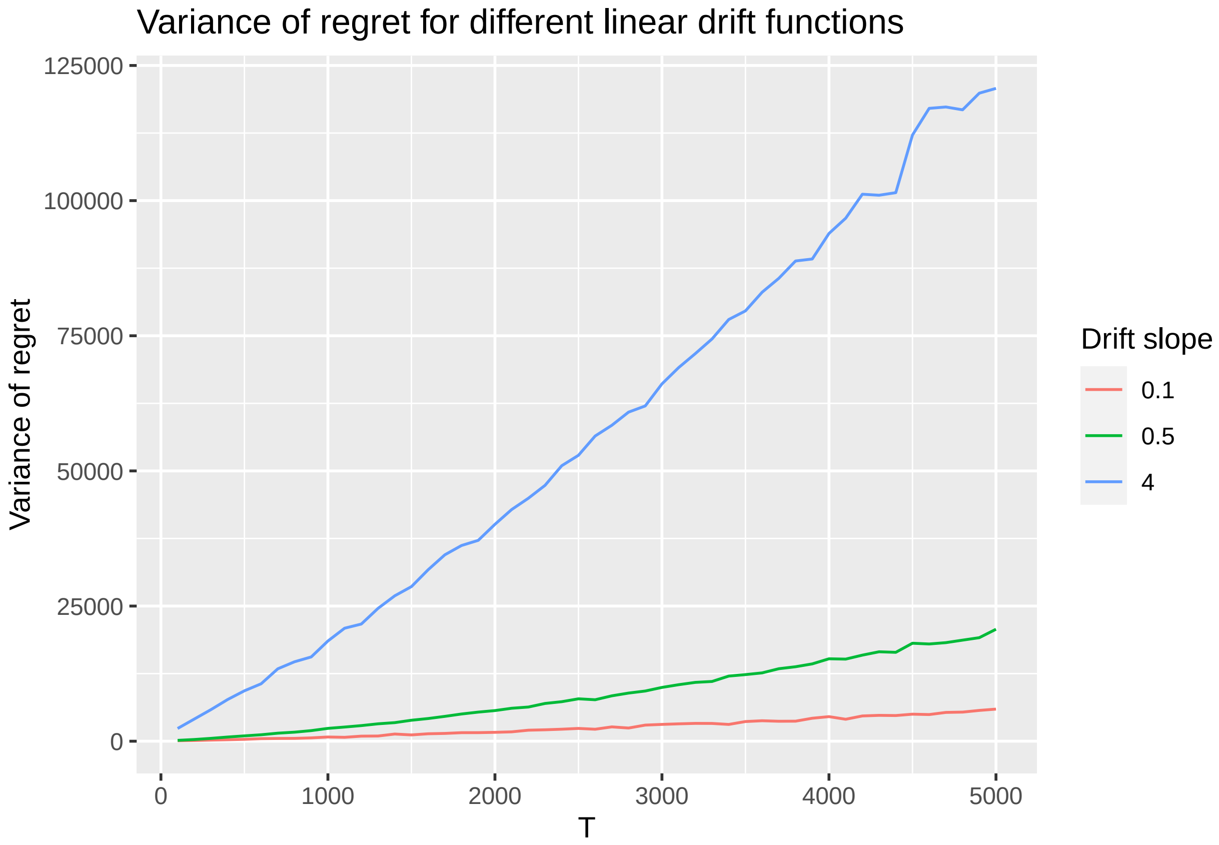
To see how the changing slope of the drift function affects the relationship between the average cumulative regret and the time horizon, we can use non-linear least squares to fit curve:

$$R = c \cdot T^p$$

```
## # A tibble: 3 x 7
##   C       pow   powSE powSig     c    cSE cSig
##   <fct> <dbl>   <dbl> <lgl>  <dbl>  <dbl> <lgl>
## 1 0.1   0.628 0.00445 TRUE   0.388 0.0140 TRUE
## 2 0.5   0.642 0.00282 TRUE   0.642 0.0147 TRUE
## 3 4     0.602 0.00362 TRUE   2.16  0.0632 TRUE
```

Here we do see a small decrease in the power of the fit when increasing the drift slope coefficient, but the greatest impact is on the constant factor c, which drastically increases with the increase in the slope coefficient of the drift function.

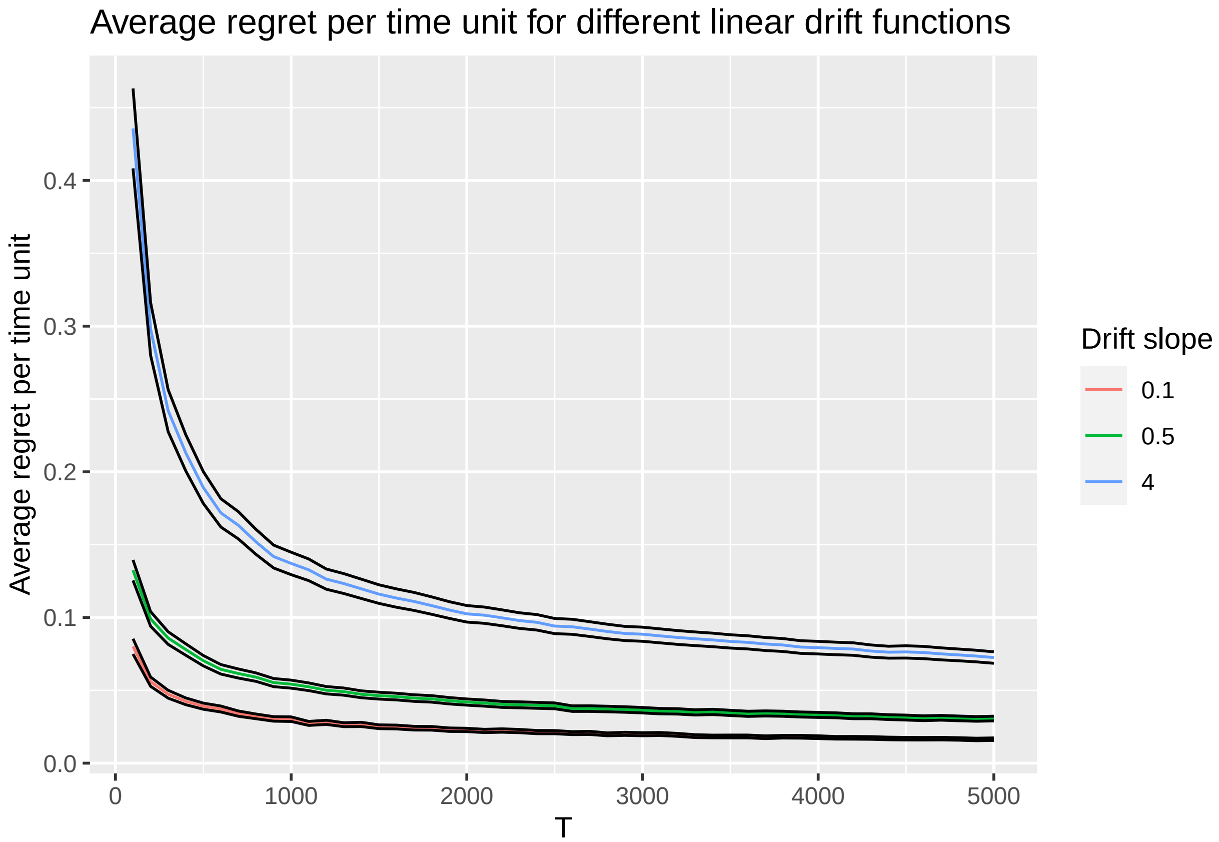
Lets look at the variability of the regret through the variance of the regret for the three different drift slopes and how it changes with increasing T.



Here we see that all three drift slopes has an increase in the variance, when T increases, but the slope of the increases is drastically different. The higher the drift slope is the higher the increase is for the variance of the regret.

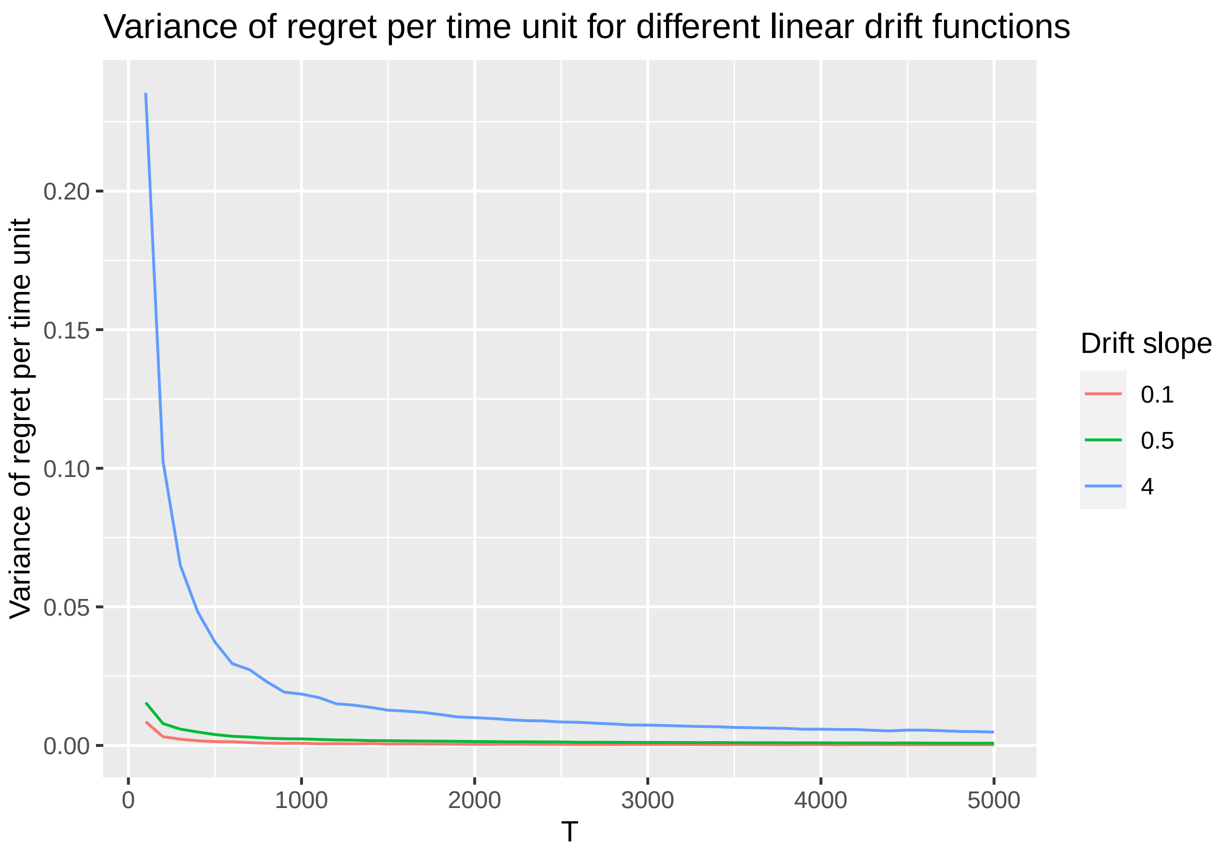
Testing how it looks for regret per time unit

To be able to compare the behavior of the algorithm over different time horizons, then it might make more sense to look at the regret per time unit, as we would expect that with more time and therefore more exploration time the data-driven algorithm would perform better and get a lower regret. This is also shown in the plot of the average regret, since the slopes are flattening as the time horizon is increasing, but it might become more evident when we look at the regret per time unit.



As expected, we see that the regret per time unit is decreasing as the time horizon is increasing. Moreover, we see that for the lower values of the drift slope, the regret per time unit isn't changing much after $T = 1000$, but for the higher drift slope the algorithm needs a time horizon of about $T = 2000$ before the regret per time unit isn't changing much.

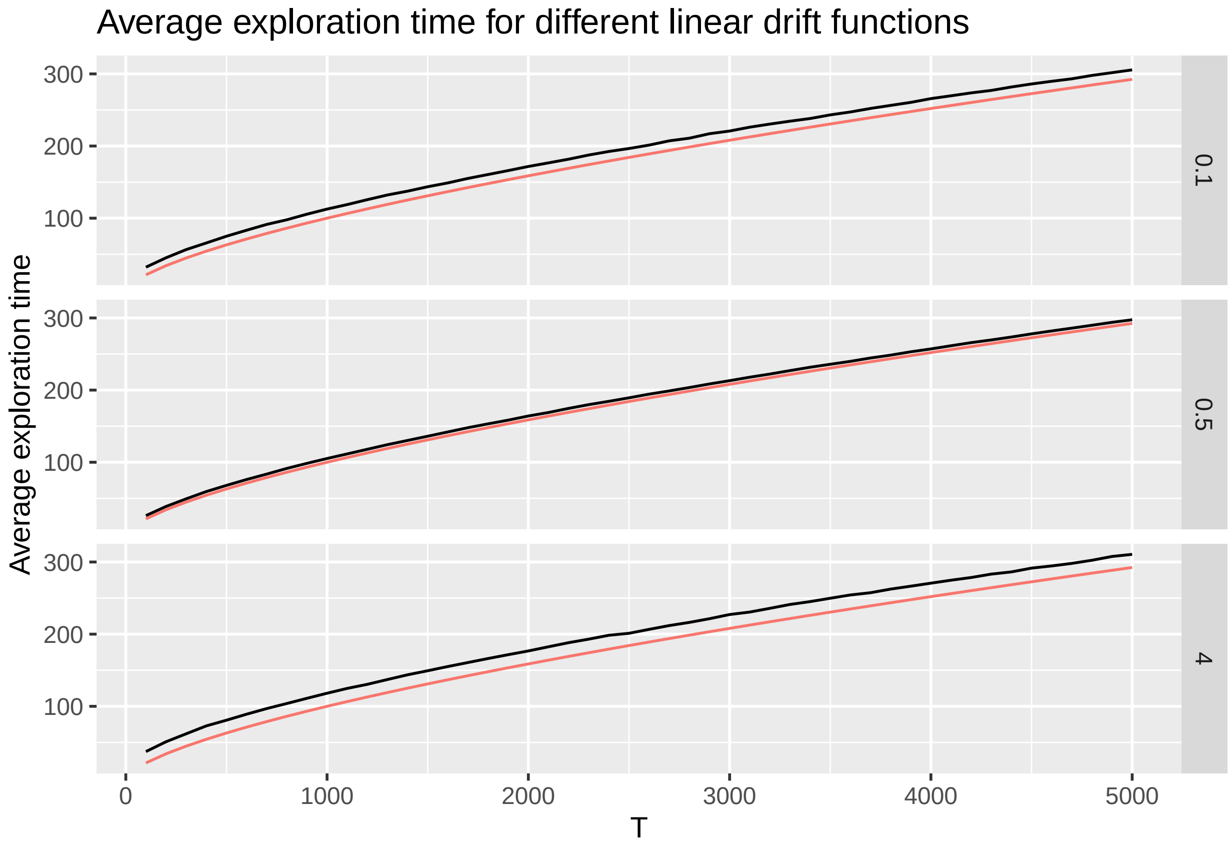
Lets also look at the variance:



Here we see that the variance of the regret per time unit is highly affected by the change in the slope of the drift function, when the time horizon is small, but as the time horizon increases, then the variance of the regret per time unit is becomes almost identical.

Exploration times

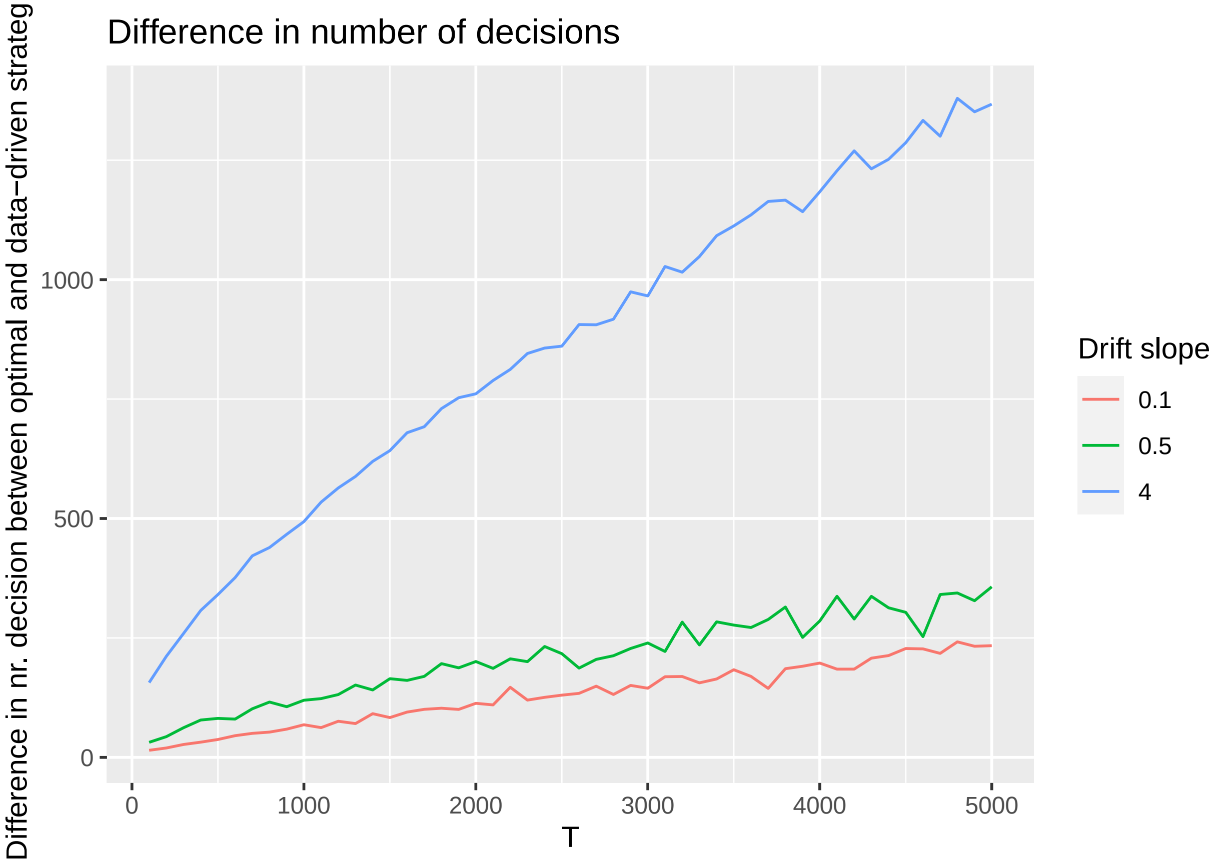
Lets confirm that the exploration times on average was of order $T^{2/3}$, and look at the confidence bounds as well, to ensure that there aren't too much variability in the exploration times affecting the regret.



It is clear that exploration times follow the theoretical exploration times very well with small deviations for both $C = 0.1$ and $C = 4$. Therefore, it does not seem like this should be causing a difference in the average regret.

Average number of decisions

Lets see how the average number of decisions are affected by the drift slope



Here it is clear that the difference in number of decisions gets larger with a larger drift slope, which therefore would result in a higher regret.

Note: Would be interesting to see how the estimated threshold distribute according to the optimal threshold. The algorithm is affected through the expected hitting times, when changing the drift, and with a higher drift slope it matters if the estimated threshold is higher or lower than the optimal threshold, since if the estimated threshold is higher than the optimal threshold, then it will take longer for the process to reach the threshold, and the optimal strategy will then be able to make more decisions than the data-driven approach.

Reward functions

Now having looked into how the drift function affects the regret for the data-driven algorithm lets see how the reward function affects the regret. We have two aspects of the reward function that we are testing. The first is the changing of the power p in the reward function, which changes the steepness of the reward function. The other is the reward functions closeness to 0 as X_t approaches 0.

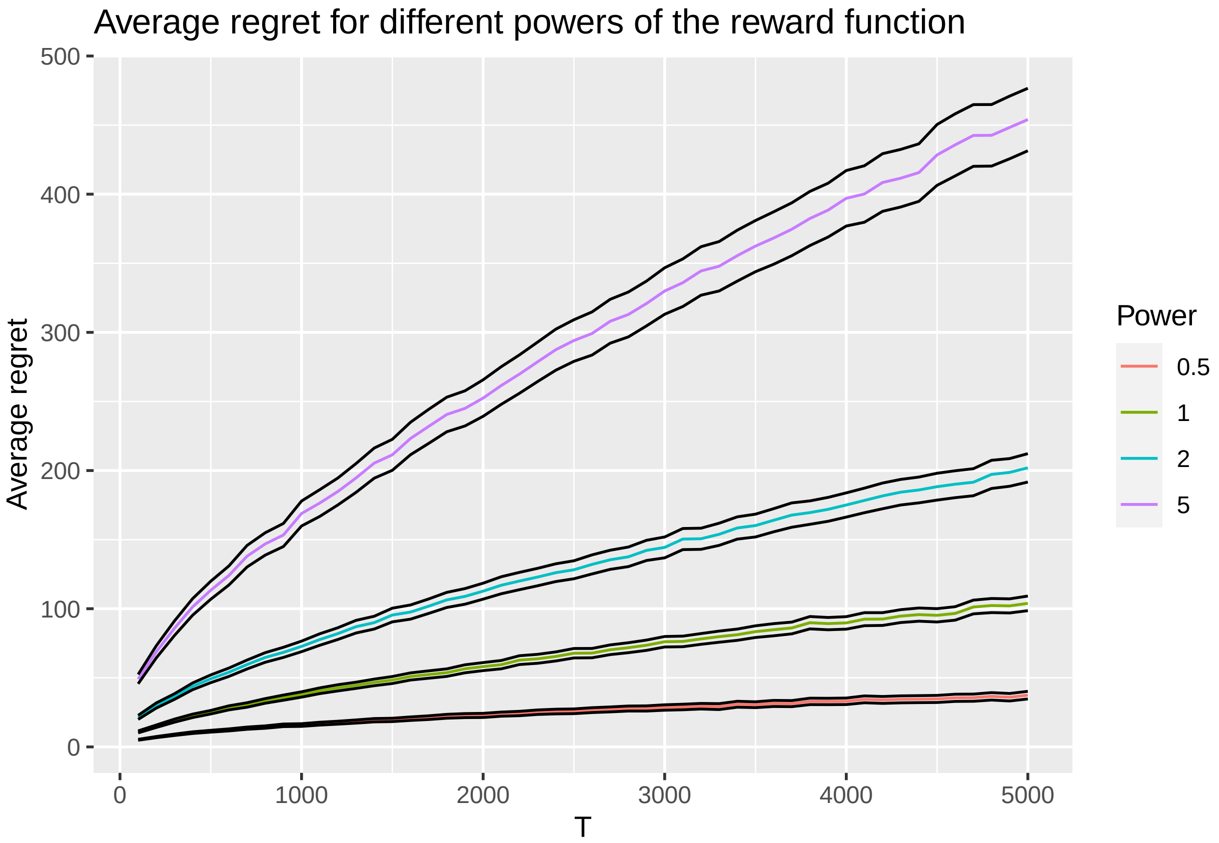
Power of the reward function

Lets first look at the steepness of the reward function and see how it affects the optimal threshold:

```
## # A tibble: 4 x 2
##   power avgOptThreshold
##   <dbl>           <dbl>
## 1   0.5           0.814
## 2   1               0.491
## 3   2               0.250
## 4   5               0.109
```

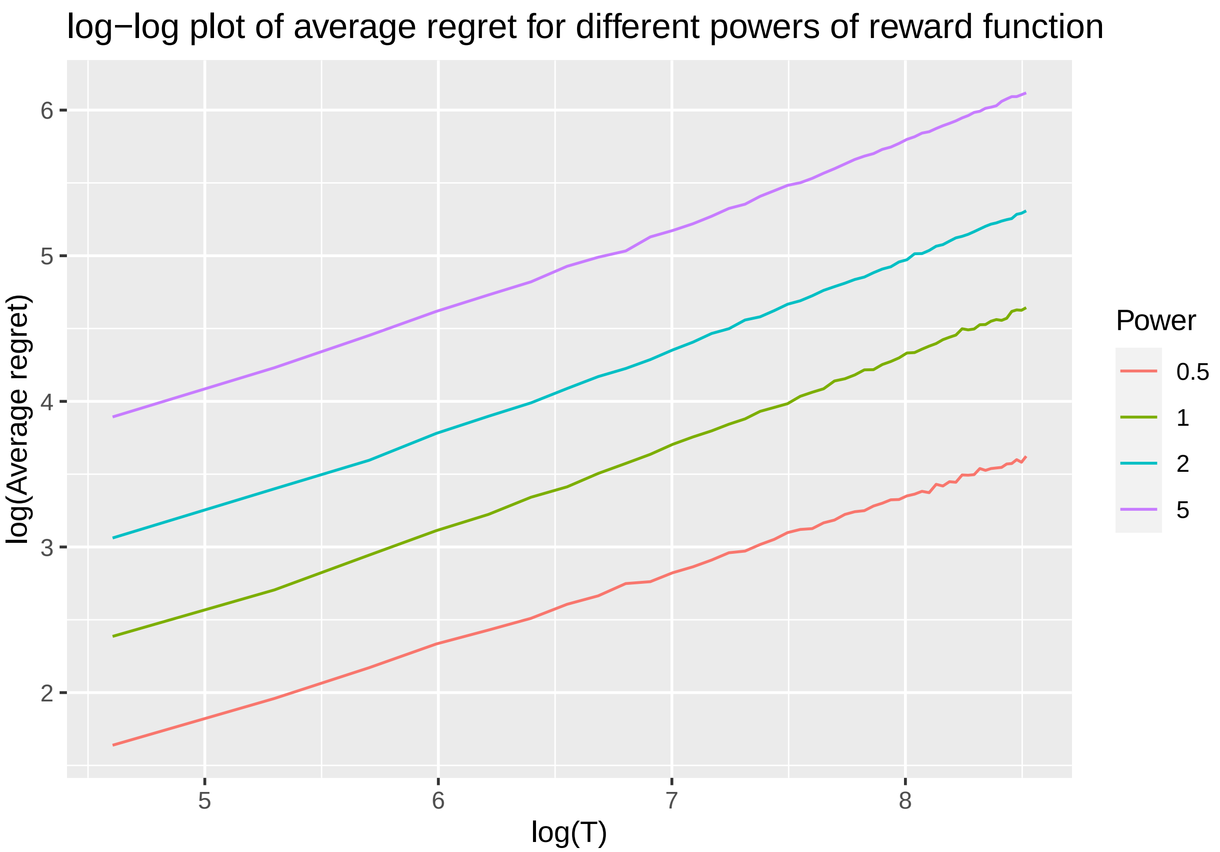
Here we see that a lower power, meaning a more steep reward function increases the optimal threshold on average, which makes sense, as a small decrease in the threshold will have a higher impact on the reward gained.

Plotting the average regret lines with their 95 % confidence interval based on the standard error for the different power values of the reward function.



Here we see that a higher power of the reward function meaning a less steep reward function increases the average regret, which makes sense, since when the reward function is less steep, then the reward function has less impact on the objective function, and the data-driven strategy has to have a more accurate estimate of the invariant density and expected hitting times in order to obtain a more precise estimate of the optimal threshold.

As before, we can use the log-log plot to get an idea of the how the power of the regret affects the rate of the average regret over the different time horizons.



Here we see that for all powers the slopes are almost the same, which would indicate that changing the power of the reward function has little to no effect on the exponent of the average regret per time, but we do see a difference in the intercept, meaning that the constant is affected by the power of the reward function.

To see this more exactly, we can look into the coefficients if we use non-linear least squares to fit the average cumulative regret by:

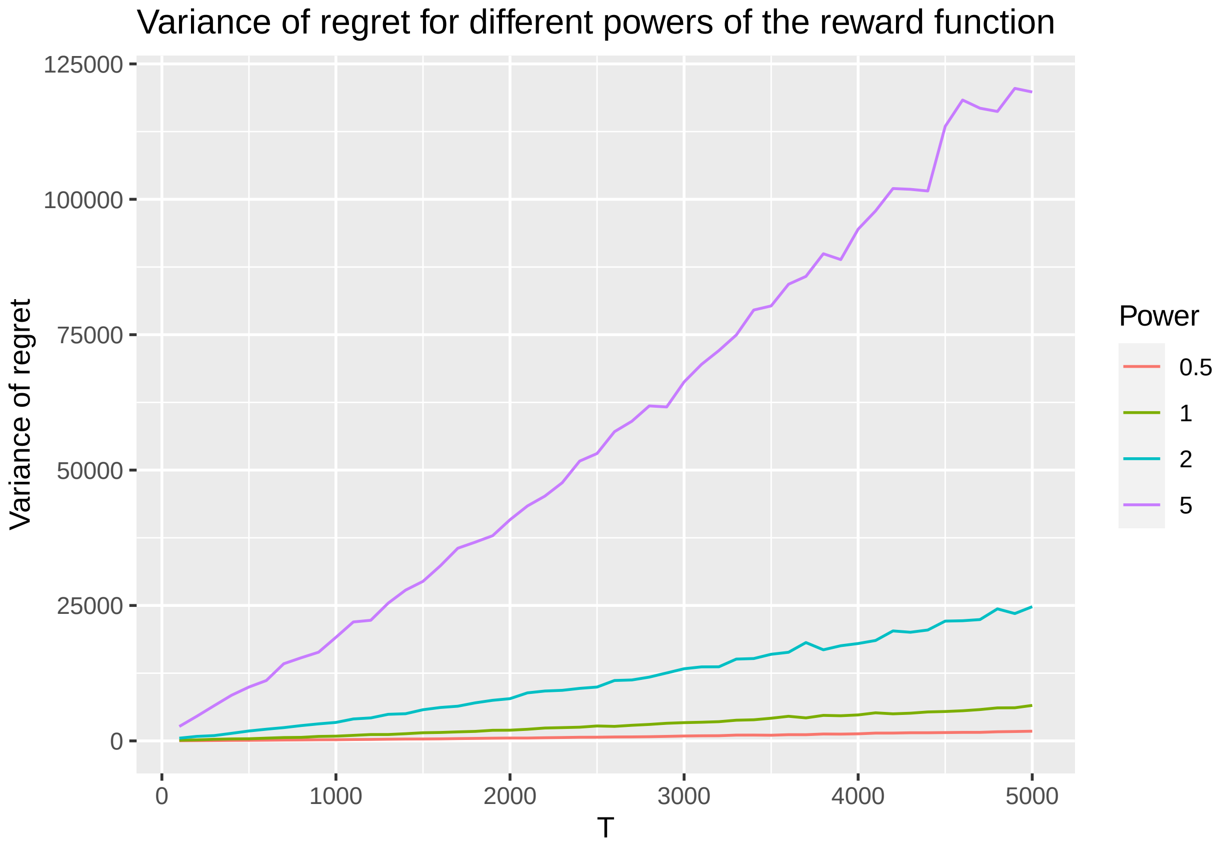
$$R = c \cdot T^p$$

```
## # A tibble: 4 x 7
##   power   pow   powSE powSig     c    cSE cSig 
##   <fct> <dbl>   <dbl> <lgl>  <dbl>  <dbl> <lgl>
## 1 0.5   0.516 0.00352 TRUE   0.457 0.0129 TRUE 
## 2 1     0.621 0.00370 TRUE   0.521 0.0156 TRUE 
## 3 2     0.623 0.00304 TRUE   0.992 0.0244 TRUE 
## 4 5     0.619 0.00334 TRUE   2.32  0.0628 TRUE
```

Here we see that the power of the fit doesn't really change for the powers $1, 2, 5$ of the reward function but that it does change when we decrease the power of the reward function to $1/2$. However, the effect isn't large, and we see that the largest effect to the fit is the increase in the constant c when we increase the the power of the reward function.

- We can consider looking at the residuals to see how well the fit actually is and if this fit makes sense.

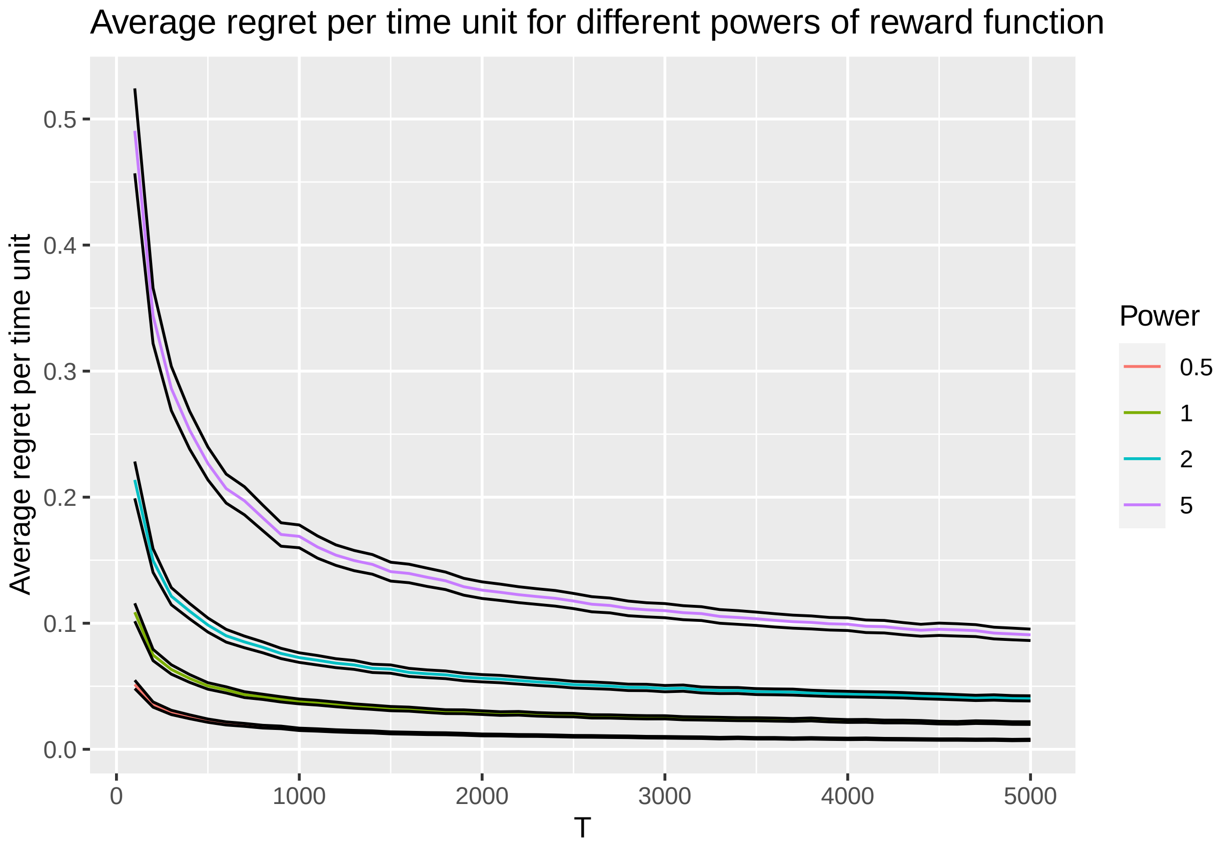
Lets see how the power of the reward functions influences the variance of the regret.



Here we see a the same effect as with the increasing drift slope, where when we increase the power of the reward function, then the variance of the regret increases and the data-driven algorithm becomes less consistent.

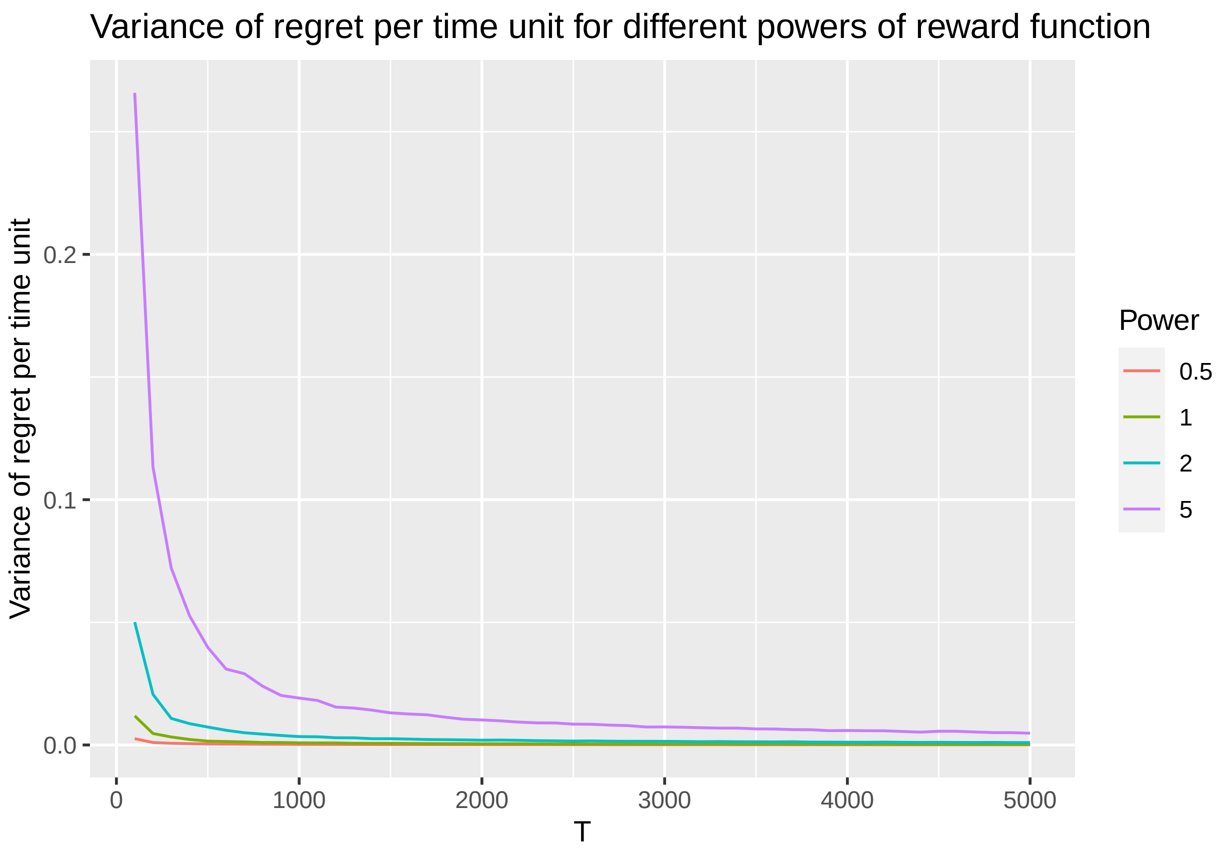
Testing how it looks for regret per time unit

As with the different drift functions, we will look into how the power of the reward function affects the regret per time unit.



As expected, we see that the regret per time unit is decreasing as the time horizon is increasing. Moreover, we see that increasing the power of the reward function also delays the point at which the average regret per time unit is stabilizing.

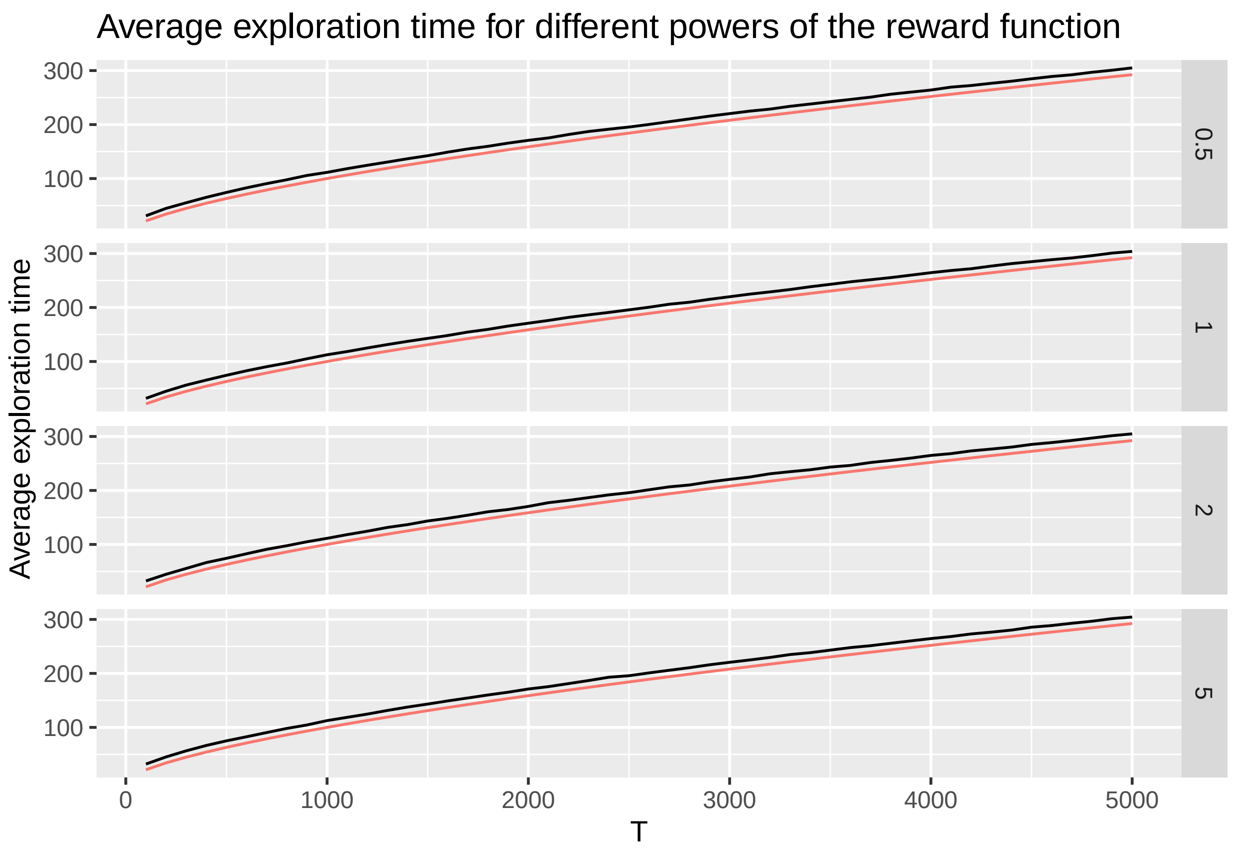
Lets also look at the variance:



Here we see that the variance of the regret per time unit is highly affected by the change in power of the reward function, when the time horizon is small, but as the time horizon increases, then the variance of the regret per time unit is becomes almost identical, as was the case with the increasing drift slope.

Exploration times

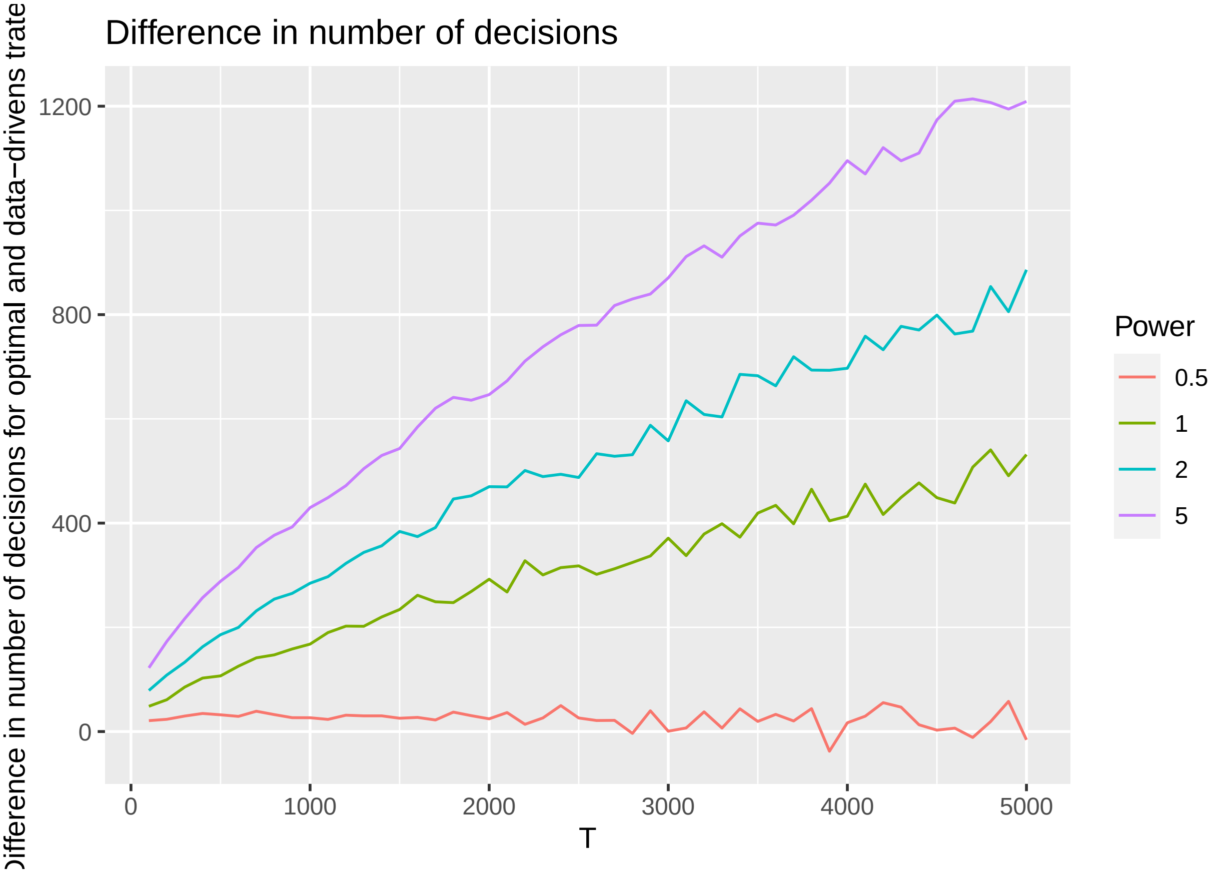
Lets again see how the exploration times distribute and if they followed the expected order.



Here we see that all follow the expected order of increase with T, so the exploration times should not be affecting the average regret.

Number of decisions

Lets see how the number of decisions changed



Here we see that increasing the power of the reward function increases the difference in number of decisions made by the optimal strategy and the data-driven strategy. Interestingly, for the power of $1/2$ the data-driven strategy is taking basically the same number of decisions as the optimal strategy, and even sometimes takes more decisions that then optimal strategy even with its exploration time.

Note: The case with $p = 1/2$ might be due to precision error of the computer, since for some drift functions and some values of a, then the optimal threshold and is close to ζ, which is difficult for the computer to find, since it is optimizing from 0 to ζ, and therefore will most likely find that the best threshold is just under the optimal threshold. Therefore, it will lead to more decisions made by the data-driven strategy.

Closeness to zero for X=0

The second variable of the reward function that we are considering is the value of a, which controls how close the reward $g(0)$ is to 0. This is considered in relation to the constraint $g(0+) < 0$, to see how robust the data-driven algorithm is to lower values of X_t being punished less, meaning that a low threshold is better. Therefore, lets see how the different values of a affects the optimal thresholds on average.

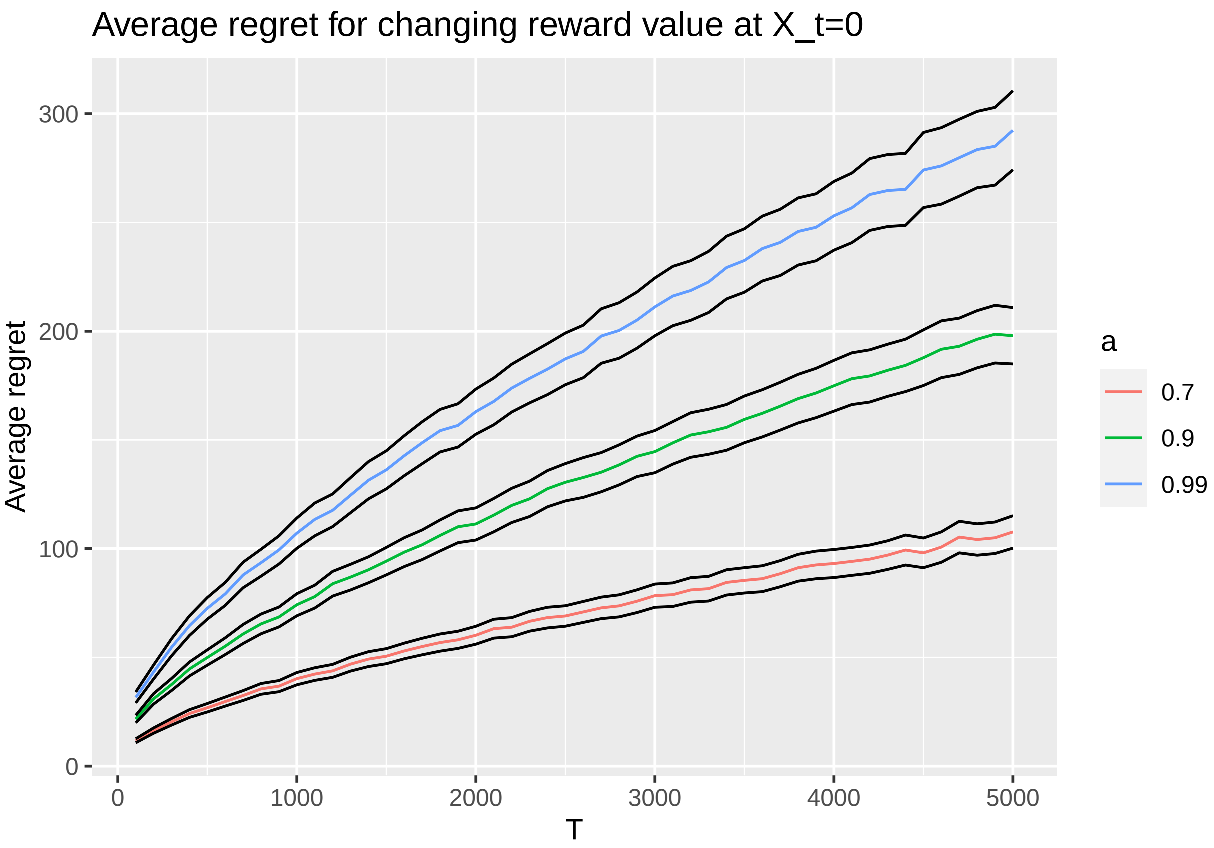
```
## # A tibble: 3 x 2
##   zeroVal avgOptThreshold
##     <dbl>           <dbl>
## 1    0.7            0.586
## 2    0.9            0.417
## 3    0.99           0.245
```

As expected the higher value of a is bringing the optimal threshold closer to 0, as lower thresholds are less punished by a negative reward. However, the value of a also has another impact that should be noted. In contrast to the power and the drift slope, then the value of a also changes the value of y_1.

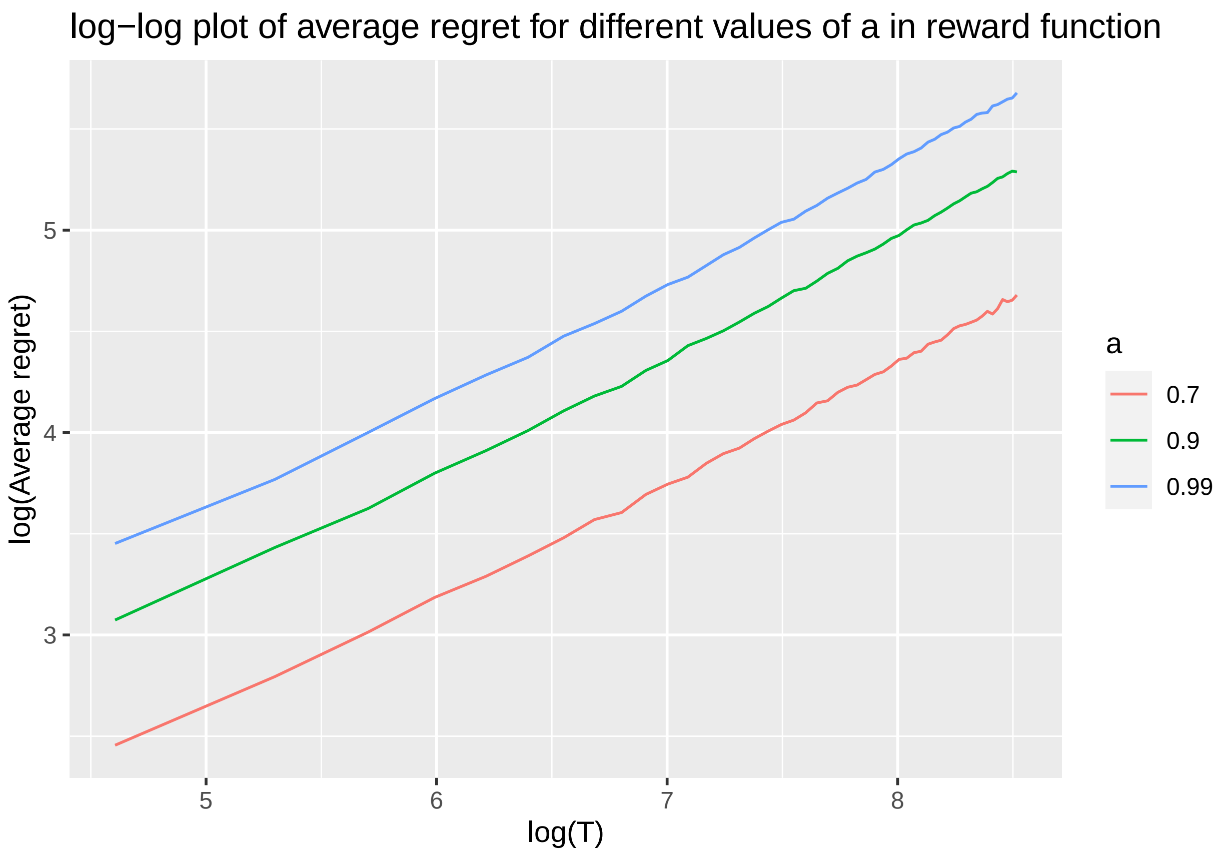
```
## # A tibble: 3 x 3
##   zeroVal  avg_y1 avg_Zeta
##     <dbl>   <dbl>    <dbl>
## 1    0.7  0.261       1.00
## 2    0.9  0.0905      1.00
## 3    0.99 0.00923     1.00
```

This affects the range of values that the data-driven is searching over when estimating the optimal threshold.

As previously, lets see how the different values of a affects the average regret:



Here we see that increasing the reward at 0 (higher value of a) does impact the average regret and that as the reward at $X_t = 0$ gets closer to 0, then the average regret increases. To get a better idea of the rate of increase we can look at the log-log plot again:



Again it does not seem to be affecting the rate that much, but it does have an effect on the constant factor.

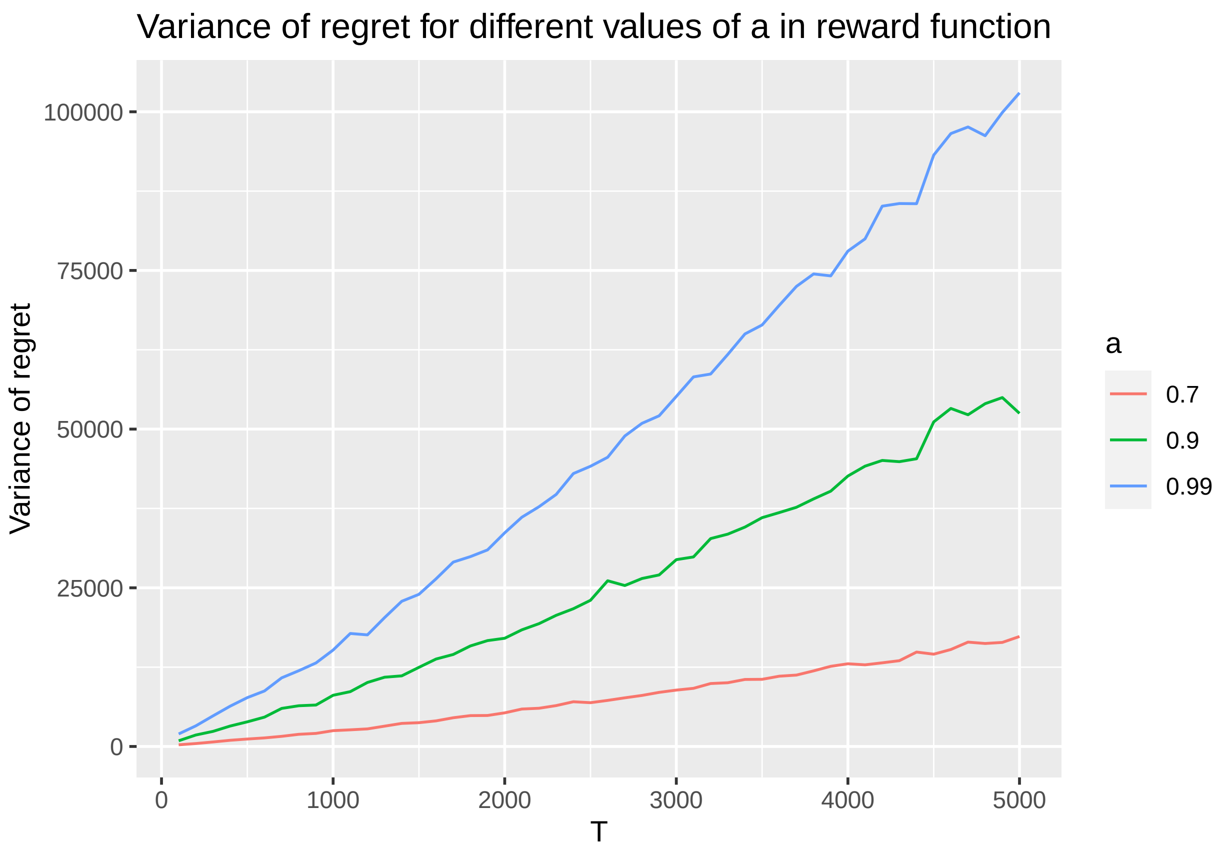
Lets again look at the values more specifically by fitting the non-linear least squares relationship:

$$R = c \cdot T^p$$

```
## # A tibble: 3 x 7
##   zeroVal   pow   powSE powSig     c     cSE cSig
##   <fct>   <dbl>   <dbl> <lgl>  <dbl>   <dbl> <lgl>
## 1 0.7     0.609 0.00374 TRUE   0.596 0.0180 TRUE
## 2 0.9     0.614 0.00307 TRUE   1.07  0.0265 TRUE
## 3 0.99    0.618 0.00307 TRUE   1.50  0.0372 TRUE
```

As expected the power of the fit does not change much with the change in the a-value, but the constant factor does.

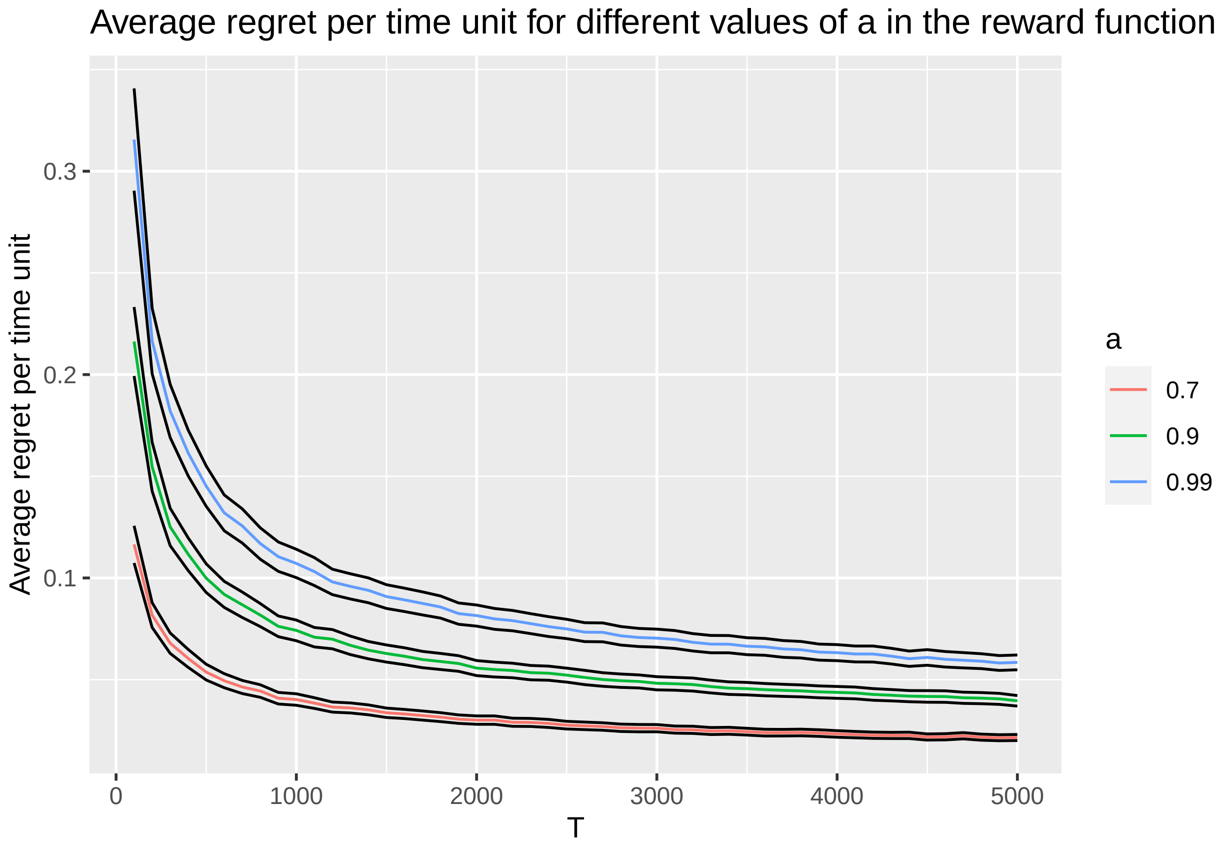
Lets look at the variance of the regret again when changing the value of a in the reward function:



Here we see that increasing the value of a, meaning that the reward for small values are less punished, increases the variance of the regret.

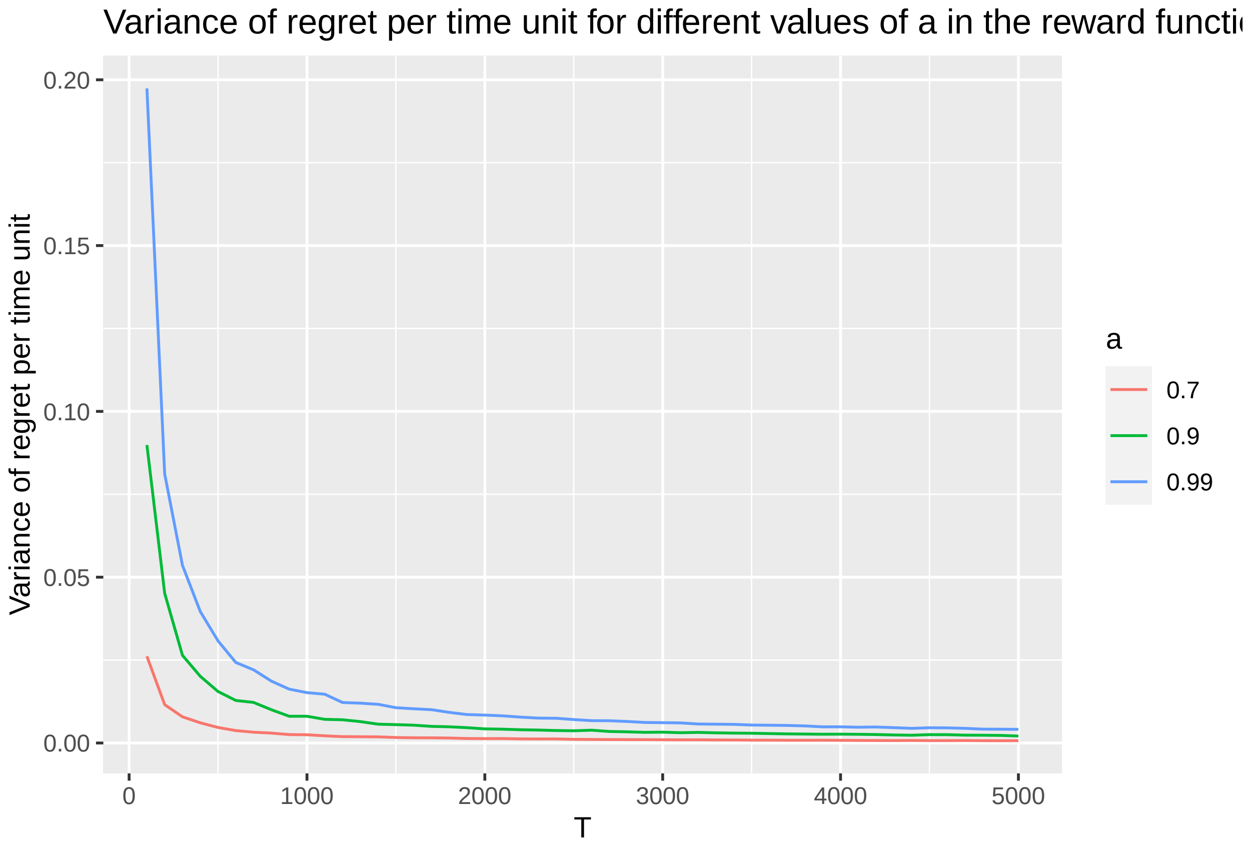
Testing how it looks for regret per time unit

As with the different drift functions and powers, we will look into how the value of a in the reward function affects the regret per time unit.



As expected, we see that the regret per time unit is decreasing as the time horizon is increasing. Moreover, we see that increasing the value of a in the reward function also delays the point at which the average regret per time unit is stabilizing.

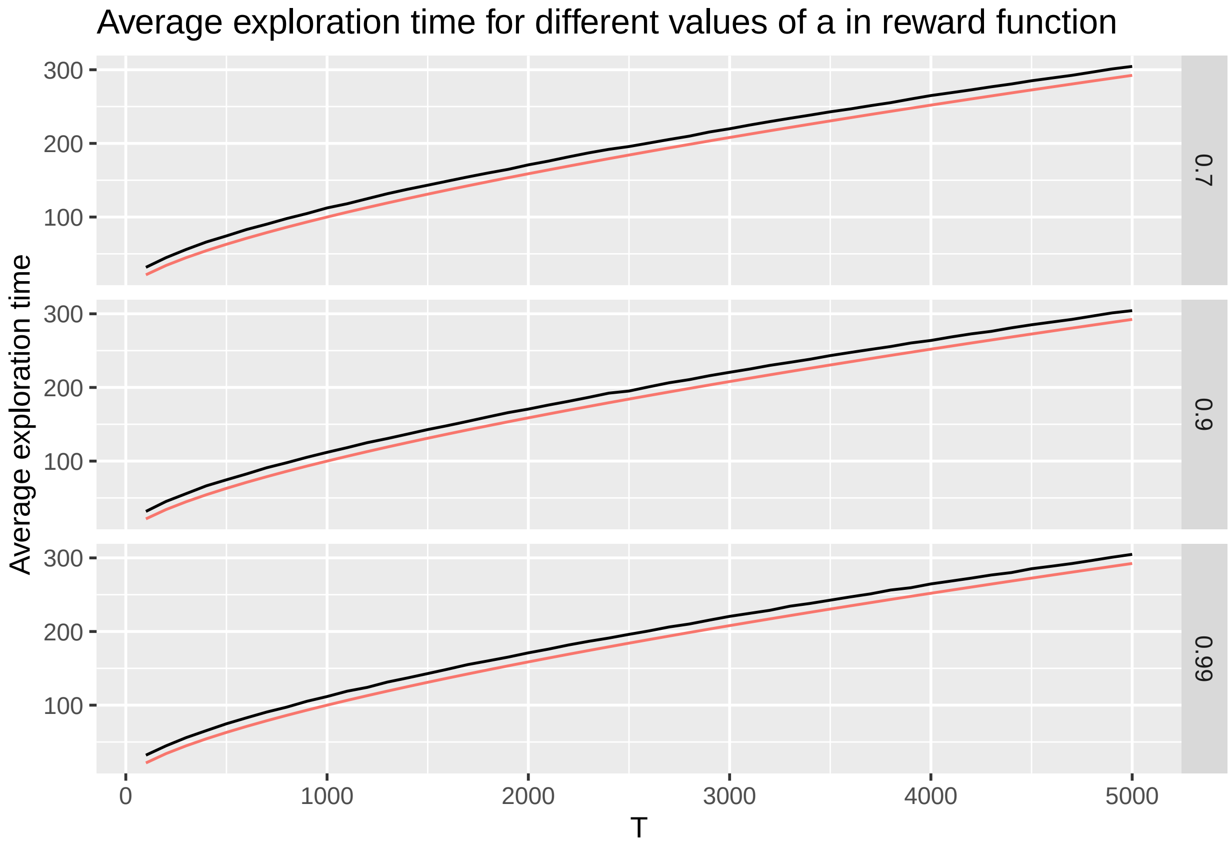
Lets also look at the variance:



Here we see that the variance of the regret per time unit is affected by the change in the value of a in the reward function, when the time horizon is small, but also that the time horizon does not have to be very long before the variance of the regret per time unit becomes almost identical, as was the case with the increasing drift slope and the power of the reward function.

Exploration times

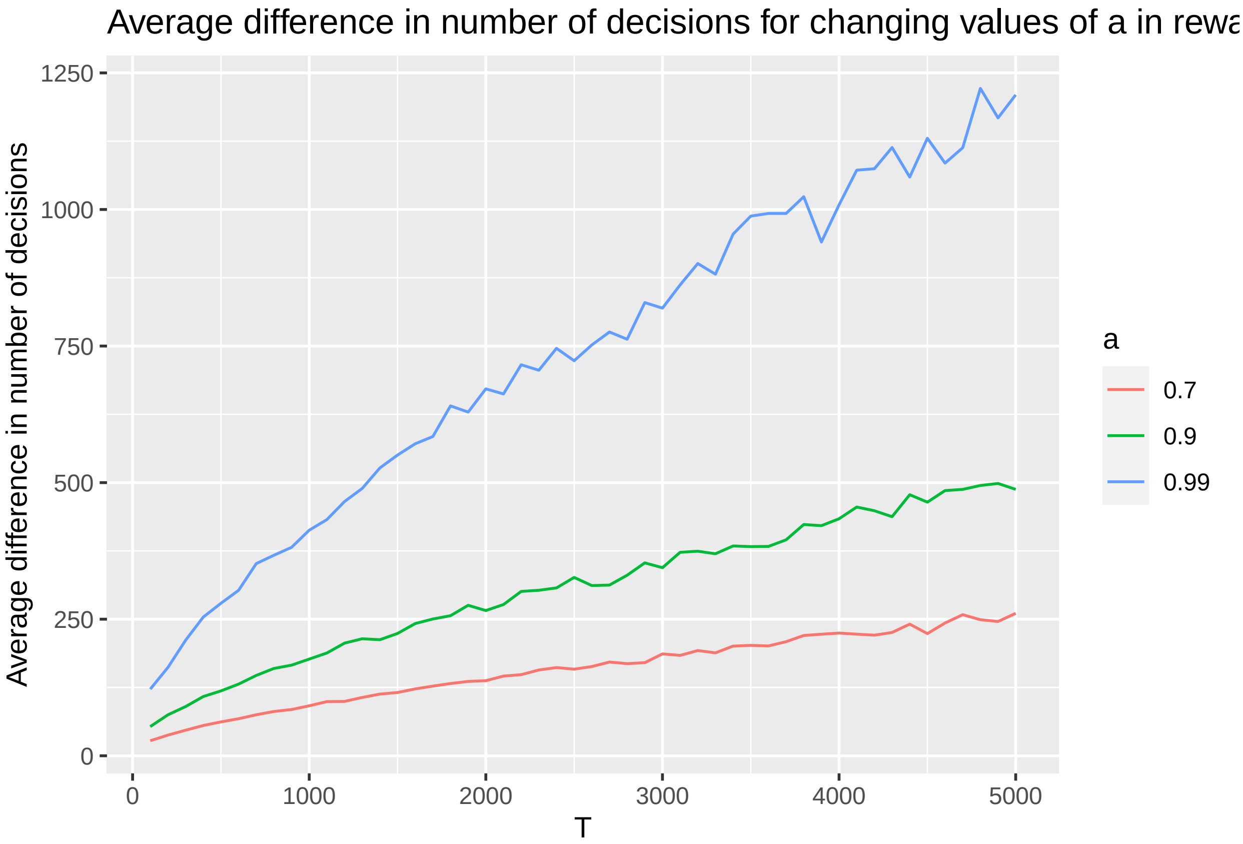
Lets check the average exploration times again



Nothing to see here either.

Number of decisions

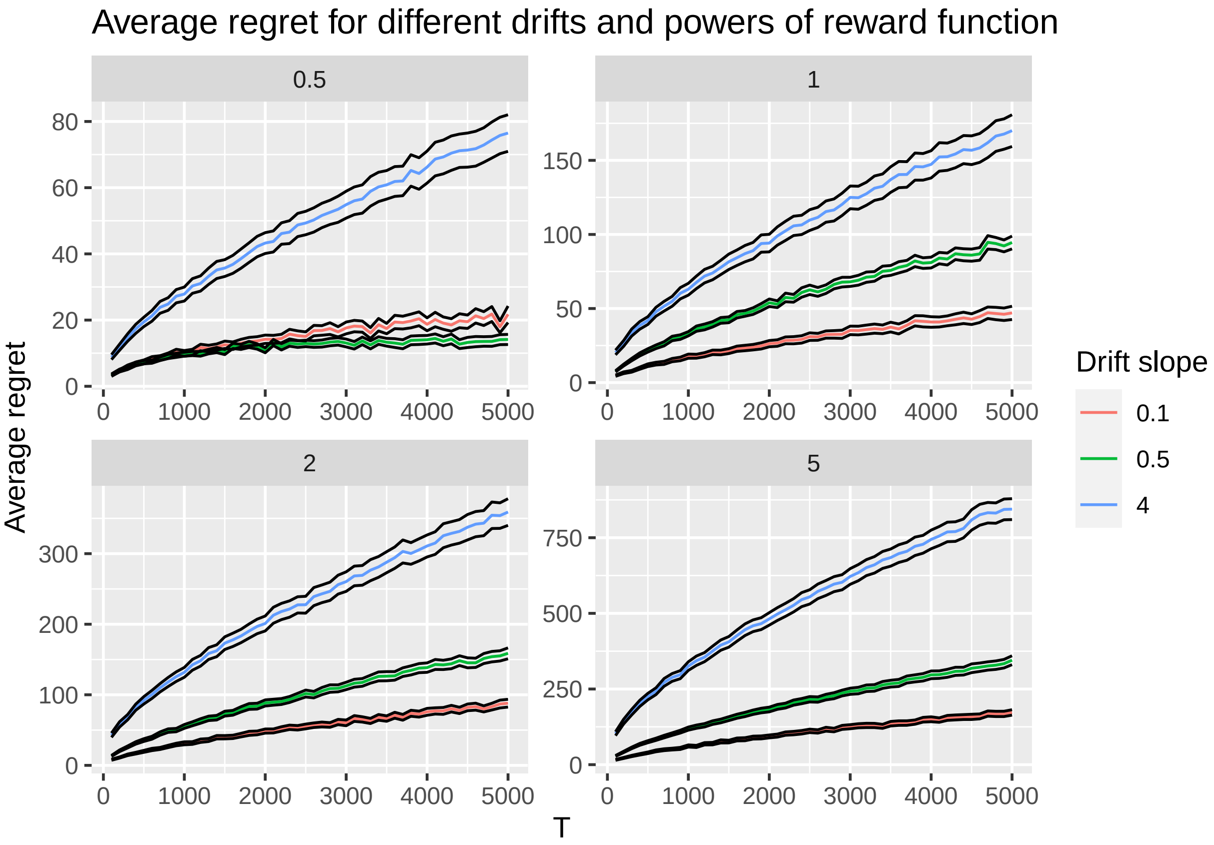
As before we will consider the difference in the number of decisions between the optimal threshold strategy and the data-driven strategy.



Here we see that a increase in the value of a does drastically change the difference in number of decisions made by the two strategies.

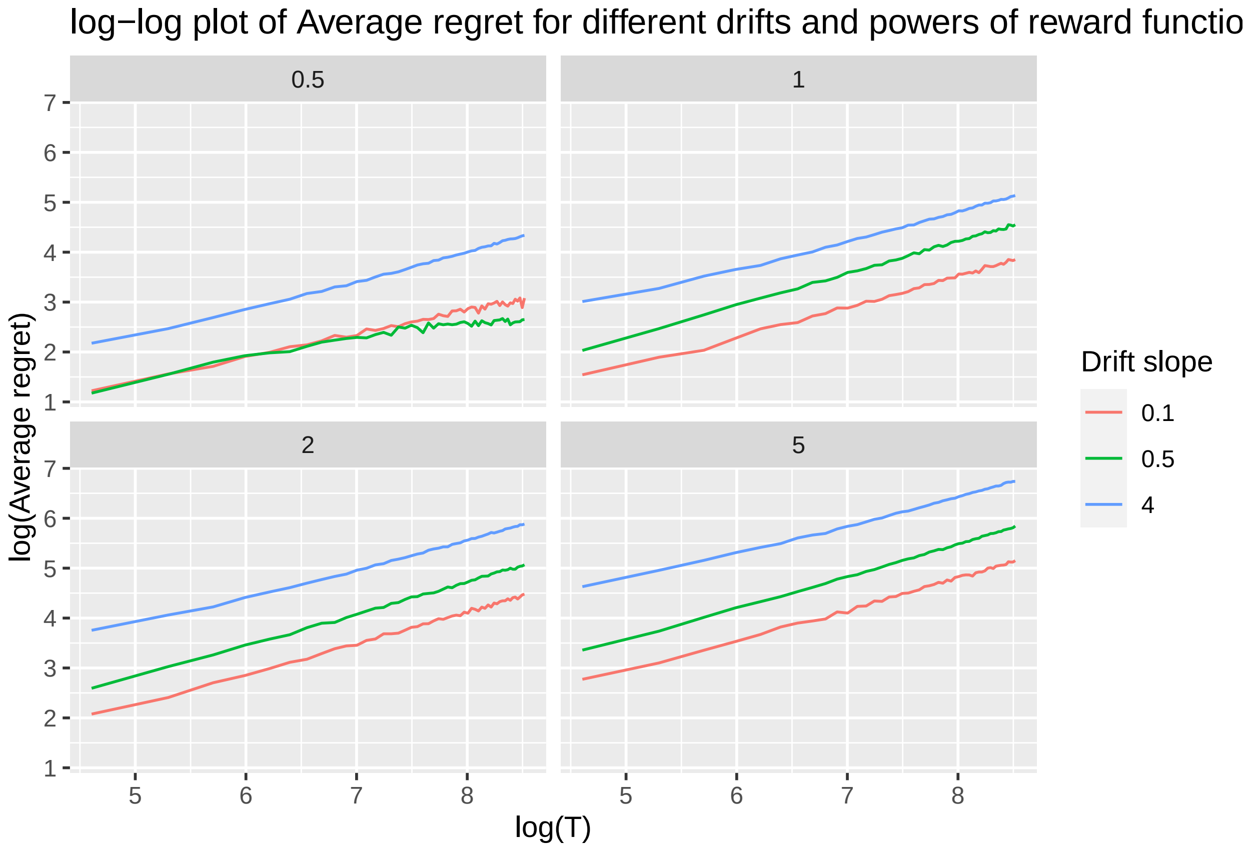
Interaction between the drift slope and reward power

Now lets see if there is any change in the average cumulative regret if we split on both the different drift functions and the different powers of the reward function. We saw previously that a lower power and a higher drift had a large impact on the average cumulative regret. Now we can see how different combinations of drift functions and powers of the reward function affect the cumulative regret.



Here we see some of the same patterns as before, where a higher slope of the drift has the highest impact on the average regret for all powers of the reward function and that the higher the power of the reward function is the more the regret is increased for all drift slopes. However, we can also see that the difference between the average regrets for the different drift slopes are increased differently depending on the power of the reward function. For a lower power, meaning a steep reward function, the high drift slope is much higher relative to the other drift slopes, that has almost the same average regret. As the power increases, and the steepness of the reward function falls it seems that a more stable results appears, where the same order of the drift slopes are consistent, but the increase between the lines are not. As the power increases, it seems that the higher drift slope is affected more than the lower drift slopes.

To see the order of increase for the average cumulative regret with increasing time horizon for the different drift functions and powers of the reward function we can look at the log-log plot again:



Here we can small changes in the slopes, but again this is not what is varying the most, and the intercept corresponding to the constant factor seems to be affected the most for varying drift functions and powers of the reward function. We see that for the low power of 1/2 the drift slope has not effect when increasing from 0.1 to 0.5, but that the increase to 4 does. Also, for this power it seems that the slightly higher drift slope of 0.5 is better for longer time horizons. For the rest of the powers we see that same order, where the lower slope coefficient for the drift function gives a lower intercept, and therefore a lower constant factor for the average cumulative regret. Also, as the power increases, so does the intercept for all of the drift functions.

To get a better idea of these values we do the same fit as before, where:

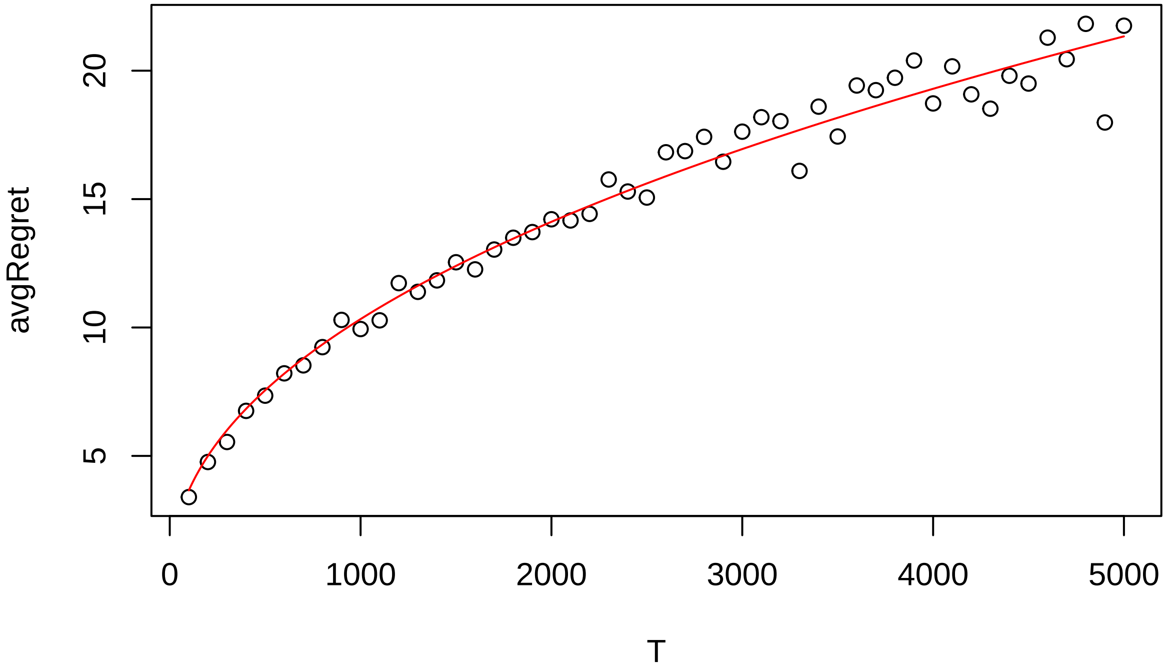
$$R = c \cdot T^p$$

	c			p		
Reward power	Drift slope = 0.1	Drift slope = 0.5	Drift slope = 4	Drift slope = 0.1	Drift slope = 0.5	Drift slop
0.5	0.459±0.0503	1.418±0.1668	0.416±0.0159	0.451±0.0137	0.275±0.0149	0.611±0
1	0.213±0.013	0.396±0.02	0.969±0.033	0.634±0.0075	0.643±0.0062	0.606±0
2	0.366±0.0227	0.62±0.0261	2.027±0.067	0.641±0.0077	0.651±0.0052	0.607±0
5	0.698±0.0365	1.208±0.0313	5.234±0.1816	0.646±0.0064	0.662±0.0032	0.597±0

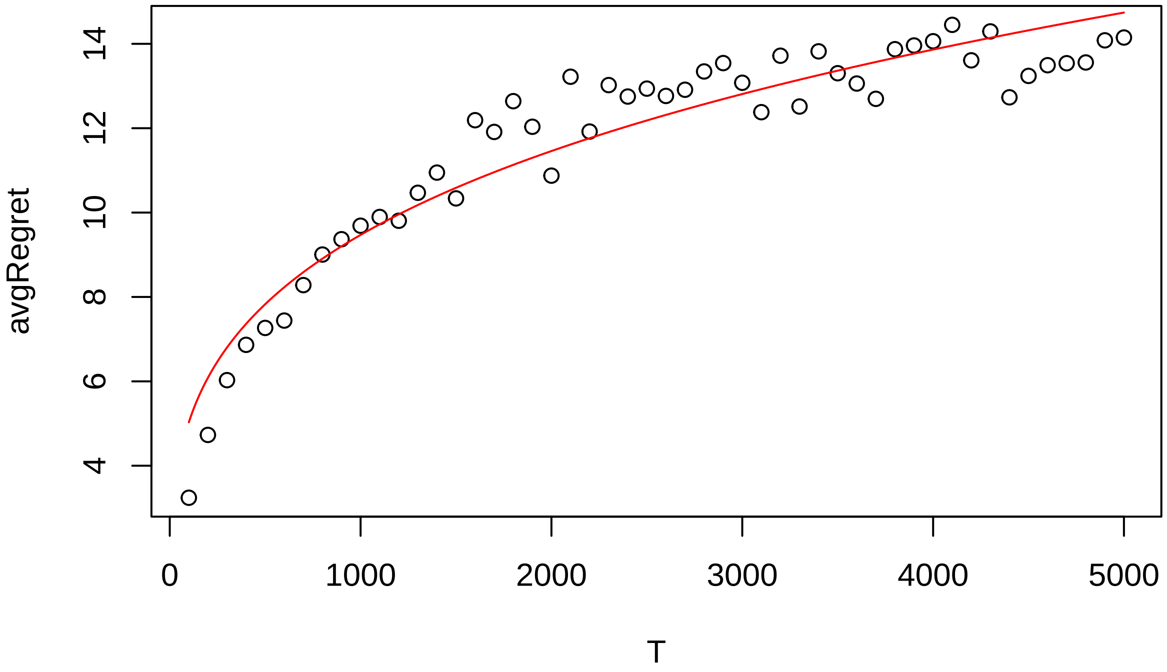
Here c is the constant factor of the fit and p is the power of the fit between the regret and the time horizon. It is clear from this table that except for the case where the drift slope is either 0.1 or 0.5 while the power of the reward function is 0.5, then the rest of the powers of the fit lies around 0.6 to 0.66, which is close to that of the upper bound. The constant c of the fit on the other hand is varying a lot and is lowest in the case where the drift slope is 0.1 and the power of the reward function is 1, while it is largest when the drift slope is high at 4 and the power of the reward is high at 5. In general we can see that the order at which the regret grows for increasing time horizons is somewhat stable, but lies in the higher order close to that of the upper-bound. However, the constant factor, which asymptotically isn't important, varies a lot for the different cases of reward and drift functions, which will have a high impact for practical applications, where one potentially only has a low time horizon.

Lets look at bit closer to the two cases, where the order of which the regret grows stood out, namely where the power of the reward function was 0.5 and the slope of the drift function was 0.1 and 0.5. We will check how the fit actually looks compared to the actual values:

First lets see the residuals in the case where $C = 0.1$:



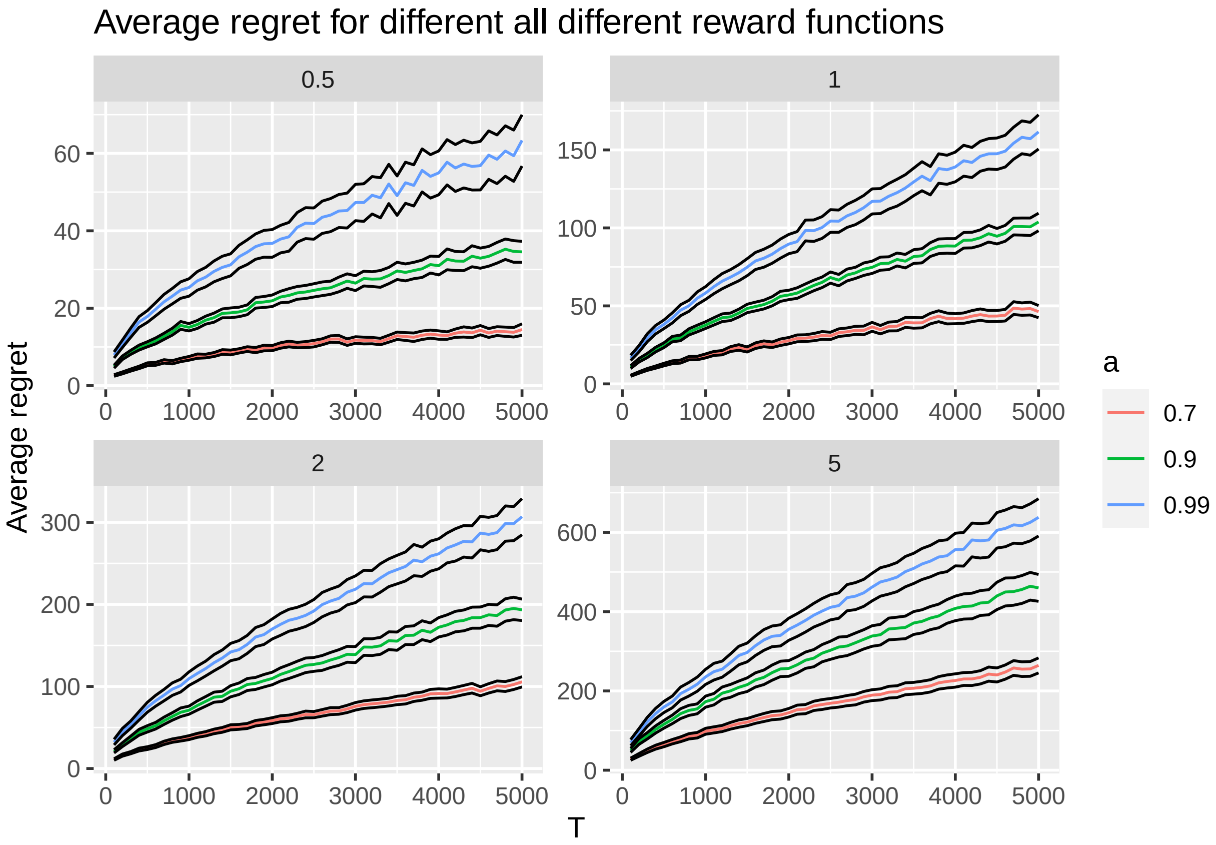
We can see that in general the fit does seem to be quite accurate, but that for longer time horizons, then the fit gets worse, since the spread of the points increase. If we look at the other case where $C = 0.5$:



We can see that the fit isn't as good, which might also indicate why for this case the estimated power of the fit was so low. It does not seem to be correct relationship between the average regret and the time horizon.

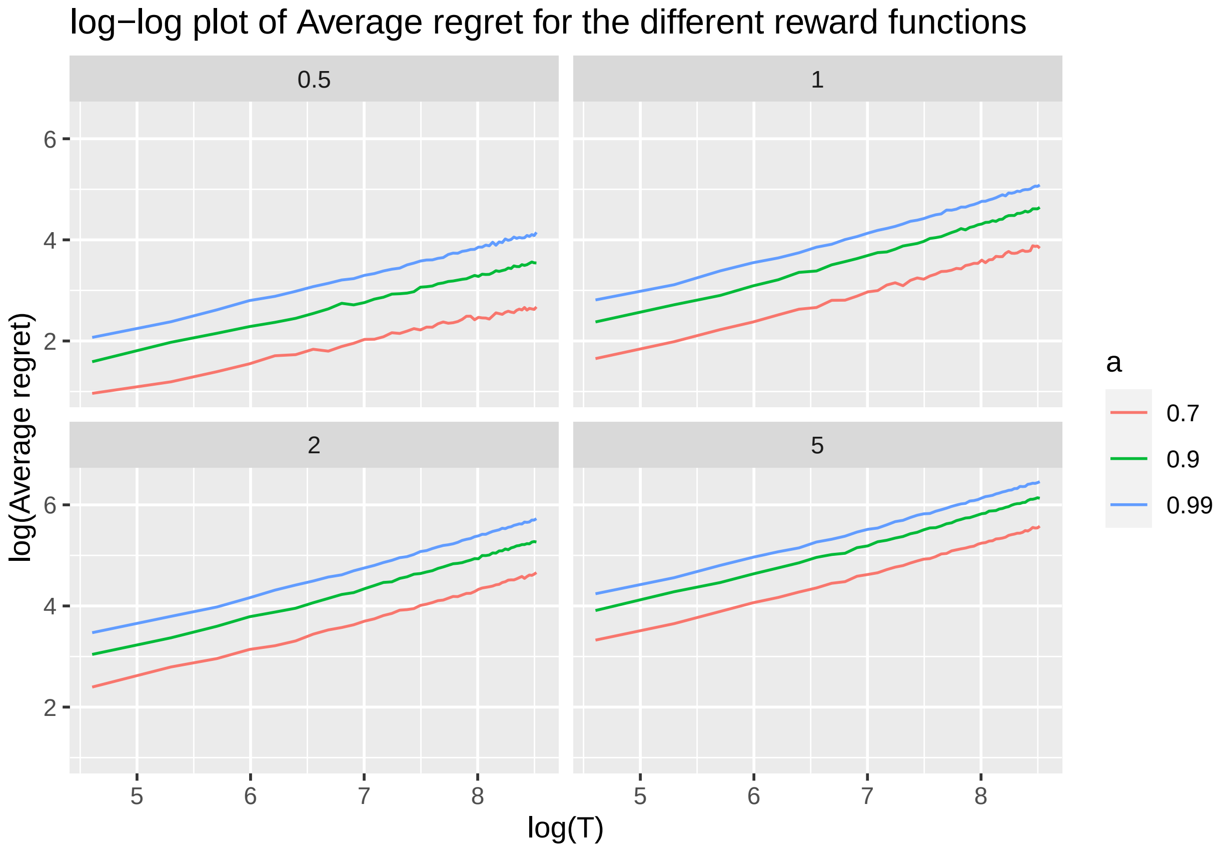
Interaction between power and a value of reward function

Lets see if there is any interaction between the power of the reward function and the a value of the reward function:



Based on this graph it does not seem that there is any interactive effects between the reward power and the value of a in the reward function. The three lines seem to behaving similar in each of the plots, and the only changes we see are the same as we saw in general for the power of the reward function and the increase in the value of a.

To make sure that this is the case we will also look at the log-log plot:



As expected the lines all seem to have the same distance to each other in all of the plots through all time horizons, and the only difference we see is the intercept for the three lines in all plots, where the difference between the lines in each plot is caused by changing the value of a, and the change in intercept for all three lines between the plots are caused by the changing power. As the relationship of the intercepts of the lines do not seem to change, then there shouldn't be any interaction between the power and value of a.

However, to make sure that this is actually the case, then we can look at the values of the fits:

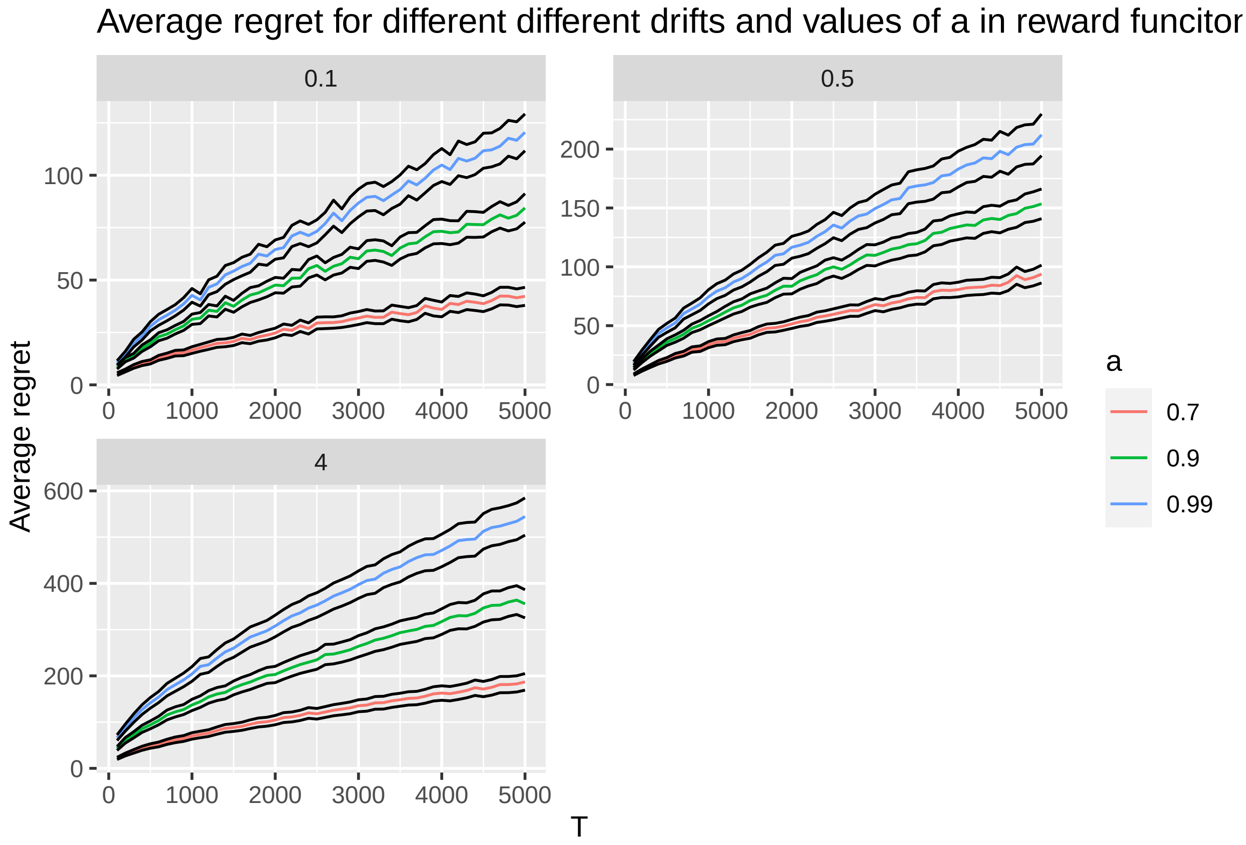
$$R = c \cdot T^p$$

	c			p		
Reward power	a = 0.7	a = 0.9	a = 0.99	a = 0.7	a = 0.9	a = 0.99
0.5	0.365±0.0241	0.458±0.0176	0.607±0.0298	0.432±0.0082	0.509±0.0048	0.543±0.0061
1	0.296±0.0209	0.491±0.0192	0.78±0.0266	0.598±0.0087	0.627±0.0048	0.625±0.0042
2	0.526±0.0216	1.029±0.0357	1.424±0.044	0.62±0.0051	0.616±0.0043	0.629±0.0038
5	1.34±0.0476	2.379±0.0798	3.251±0.1044	0.618±0.0044	0.619±0.0041	0.619±0.004

Here we do see that increasing the power of the reward function has a slightly different effect on the three different values of a and as the increase or decrease between the values for the three different values of a isn't consistent, then it must mean that there is a small interactive effect between the power of the reward function and the value of a.

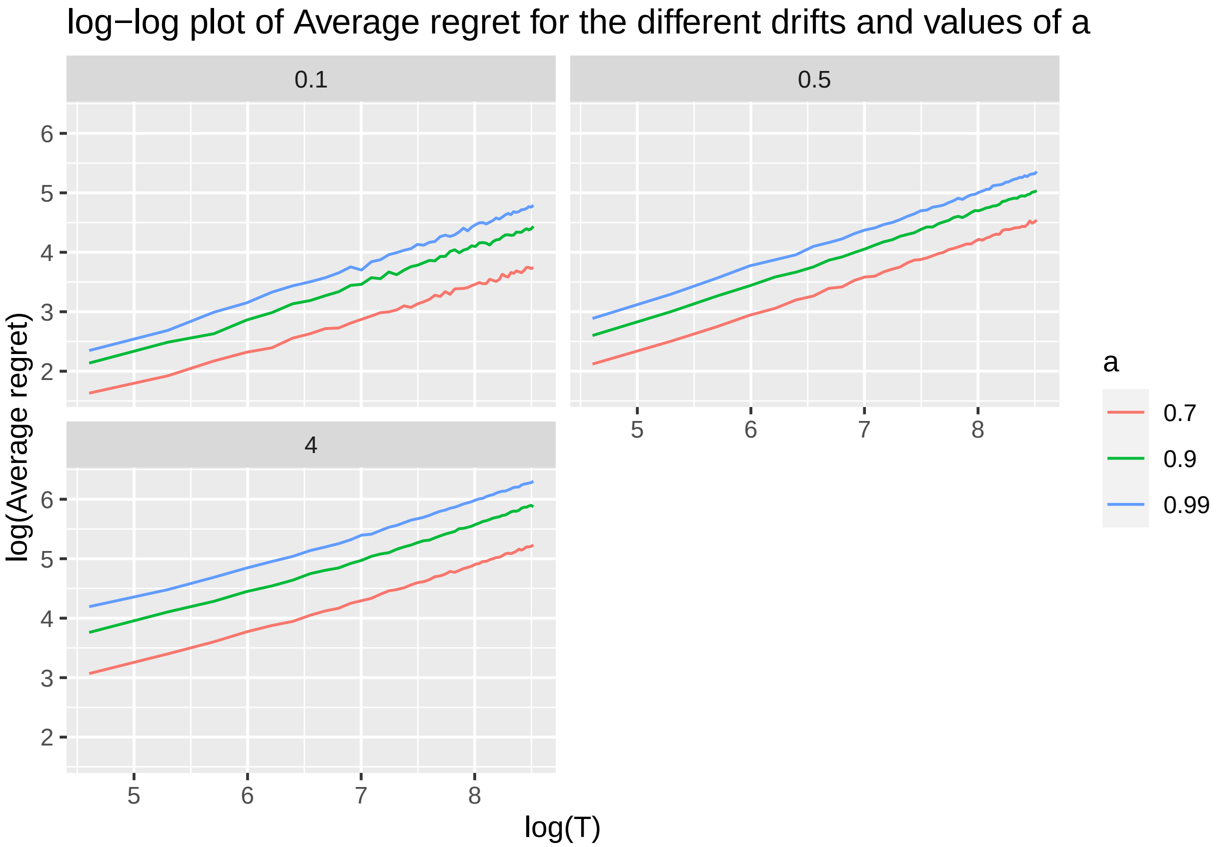
Interaction between the drift and the value of a in reward function

Lets see if there is any interaction between the power of the reward function and the *a* value of the reward function:



Here we see the same behavior as with the different slope coefficients and different values of a, but it does not seem that there are much change in the behavior if the three lines for the three different graphs.

To make sure that this is the case we will also look at the log-log plot:



It seems that the gap between the lines is slightly increasing when going from low values of the drift coefficient to high values of the drift coefficient, which would indicate an interaction that affects the constant factor in the relationship between the average cumulative regret and the time horizon, but the slope of the lines seem to be consistant between the three plots.

However, to make sure that this is actually the case, then we can look at the values of the fits:

$$R = c \cdot T^p$$

	c			p		
Drift slope	a = 0.7	a = 0.9	a = 0.99	a = 0.7	a = 0.9	a = 0.99
0.1	0.311±0.0192	0.438±0.0252	0.447±0.0234	0.577±0.0077	0.616±0.0071	0.656±0.0065
0.5	0.42±0.0182	0.674±0.0211	0.833±0.0273	0.634±0.0054	0.637±0.0039	0.649±0.004
4	1.075±0.0398	2.108±0.0847	3.298±0.0963	0.604±0.0046	0.604±0.005	0.599±0.0036

Here we do see that for the constant factor c the change for the three values of a isn't the same for all drift slope values. The change increases much more for higher values of the drift slope than for lower values of the drift slope. Also, there seem to be slight interactive effect on the power of the fit also, but in this case we have the that increase in the power from going from $a = 0.7$ to $a = 0.99$ is greatest for lower values of the drift slope.

Fit to regret table

This is just all of the individual fits for all combinations of the power and a and the drift slope coefficient. Where the same fit was used, and where “pow” is the p in $R = c{\cdot}T^p$. The big “C” is the drift slope coefficient, power is the power of the reward function, and “zeroVal” is the value of a in the reward function.

```
## # A tibble: 36 x 9
##        C power zeroVal   pow   powSE powSig     c    cSE cSig
##    <dbl> <dbl>   <dbl> <dbl>   <dbl> <lgl>  <dbl>  <dbl> <lgl>
##  1   0.5   0.5    0.99 0.118 0.0320  TRUE   4.07  1.01   TRUE
##  2   0.5   0.5    0.7  0.270 0.0177  TRUE   1.09  0.153  TRUE
##  3   0.1   0.5    0.7  0.279 0.0184  TRUE   0.763 0.111  TRUE
##  4   0.1   1      0.7  0.290 0.0278  TRUE   0.857 0.189  TRUE
##  5   0.1   0.5    0.9  0.366 0.0162  TRUE   0.765 0.0990 TRUE
##  6   0.5   0.5    0.9  0.401 0.0111  TRUE   0.742 0.0659 TRUE
##  7   0.1   0.5    0.99 0.548 0.0201  TRUE   0.365 0.0590 TRUE
##  8   4     5      0.99 0.592 0.00429 TRUE   7.74  0.268  TRUE
##  9   4     5      0.7  0.601 0.00500 TRUE   2.86  0.115  TRUE
## 10   4     5      0.9  0.604 0.00651 TRUE   5.12  0.270  TRUE
## # i 26 more rows
```