JOURNAL TITLE

Published DATE

Submitted DATE

Some methods for analyzing and correcting for spatial autocorrelation

Emil O. W. Kirkegaard

Abstract
A method for examining spatial autocorrelation based on distance scores was examined and found to be deficient. Because of this an alternative measure was proposed: k nearest spatial neighbor regression. Methods were examined across simulated datasets to see if they could successfully remove SAC and distinguish between a true and a spurious cause. Furthermore, the methods were compared to a more traditional measure of spatial autocorrelation, Moran's I.

Key words: spatial autocorrelation, Moran's I, correlation of distances, k nearest spatial neighbor, knsn

1. Introduction

Much research analyzes data from countries, states, weather stations or other units have that a location. Spatial autocorrelation (SAC) is when there are spatial patterns in the dataset (Hassall & Sherratt, 2011; Radil, 2011). This can be both positive (nearby cases are similar), neutral (neighbor cases have no particular relationship), or negative (nearby cases are dissimilar). Figure 1 illustrates this.

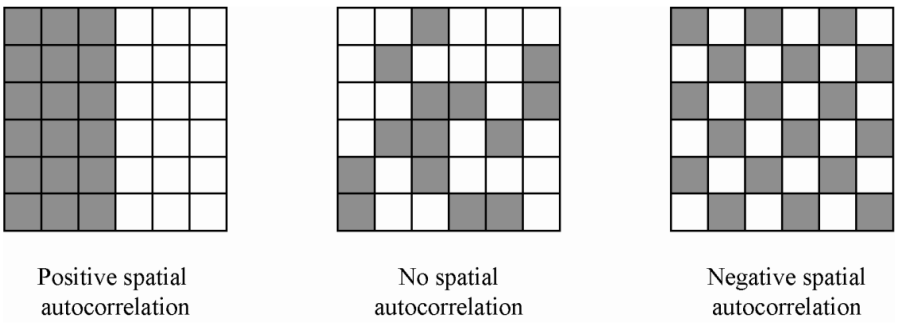

Figure 1: Illustrations of spatial autocorrelation. From (Radil, 2011).

The presence of SAC in a dataset means that the cases are not independent which means that the degrees of freedom are overestimated. There are methods for correcting the degrees of freedom, but

these are not the focus of this paper (Gelade, 2008; Hassall & Sherratt, 2011). Instead, the purpose of this article is to explore ways to examine and correct for SAC.

2. Correlation of distances method

A conceptually simple method for examining SAC the correlation of distances method (CD). CD was used by Davide Piffer (Piffer, 2015) but is so simple that it would be surprising if it had not been thought of before. The first step is to calculate distance scores for all cases for all variables of interest. Note that the kind of distance measure depends on the type of variable. Piffer analyzed, among other things, phylogenetic distances (genetic distance between two organisms or classes of organisms) and used Fst values. However, if one's data concern e.g. countries, one would use spherical geometric distances. If one is analyzing data from flatland (of any hyperspace with flat planes), one could would euclidean distances. For simple metric type variables, one uses the absolute difference. The distance scores results in a new dataset which has N(N-1)/2 cases (choose 2 of N, order irrelevant), where N is the number of cases in the original dataset.

The idea behind the method is that given two variables are linearly related, pairs of cases that are far from each other on one variable should also be far from each other on another variable. To better understand this, it is worth looking at some example data. Figure 2 shows the spatial distribution of some datapoints in flatland. An outcome variable also exists and is shown by color coding.

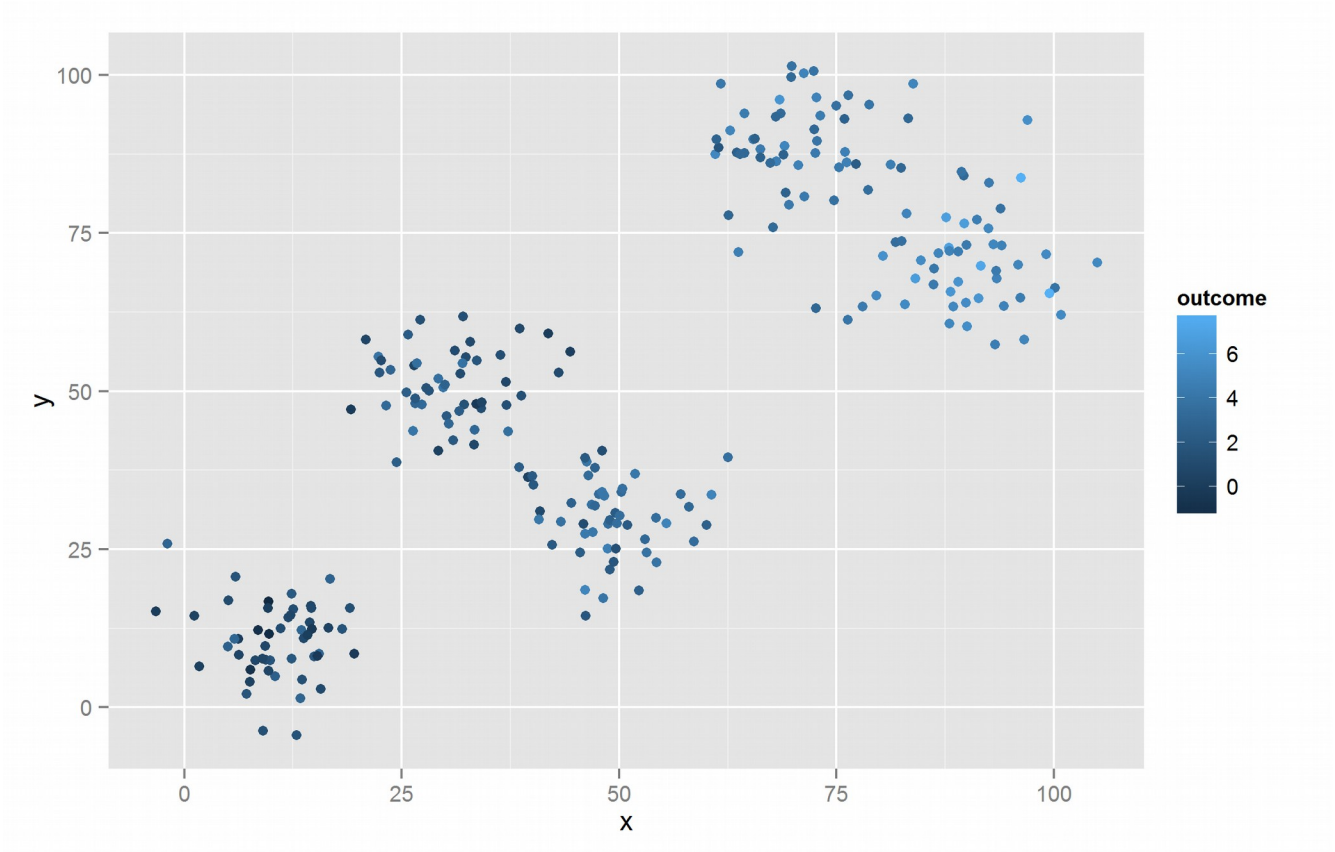

Figure 2: Dataset 1. Flatland and outcome.

There are clear spatial patterns in the data: 1) we see 5 fairly distinct clusters, 2) neighbors tend to have similar outcomes, and 3) points with higher x and y values tend to have higher outcomes.

Using the distance scores, we can examine (2) with a standard Pearson correlation, as shown in Figure 3.

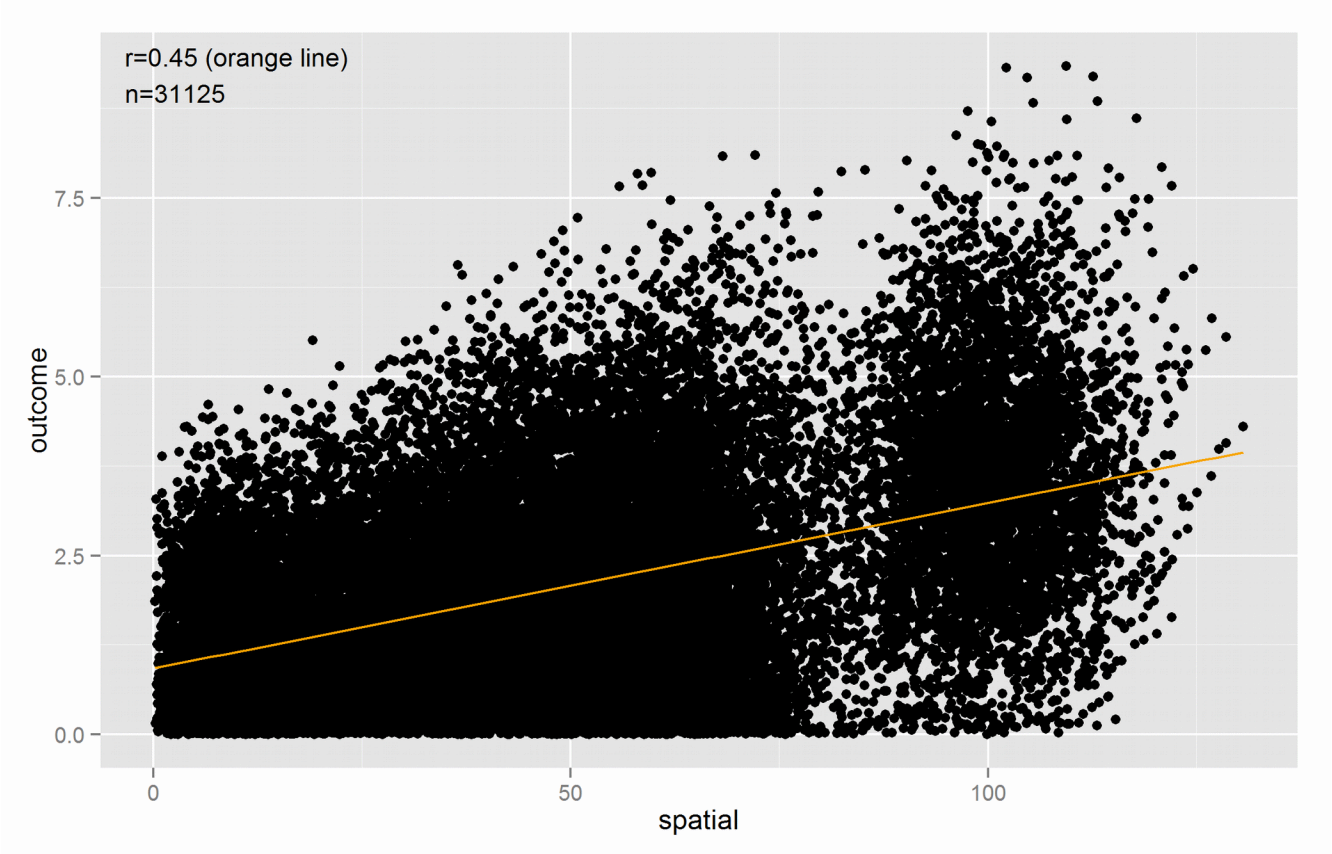

Figure 3: Scatter plot of distance scores for spatial and outcome.

As expected we see a positive correlation, i.e. cases closer to each other do have more similar outcomes. We also see substantial non-linearity in the plot. This is because, I think, all case pairs are used, making this a kind of global measure of SAC.

If we want to control for SAC, we can use standard statistical methods such as multiple regression or semi-partial correlations. Suppose we have a predictor that is spatially distributed as shown in Figure 4.

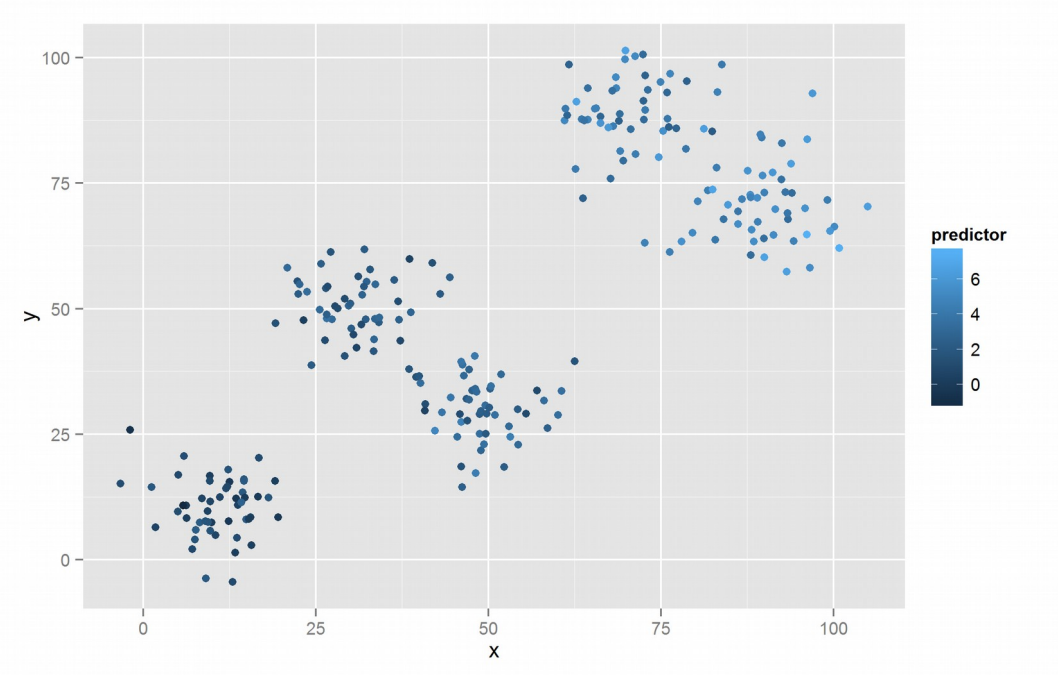

Figure 4: Dataset 1. Flatland and predictor.

We see that there is some similarity to Figure 2, but it is not immediately obvious whether the relationship owes to a pattern at the cluster level or whether the pattern is general. In the original data the correlation between predictor and outcome is .66 and using CD it is .36.

The semi-partial correlation between predictor and outcome is .13, a fair bit lower than .36, which gives us some evidence that the relationship owes much of its size to the cluster pattern. If we instead use multiple regression the standardized betas are .16 and .36 for predictor and spatial distance, respectively. In fact, the true correlation of predictor and outcome once SAC is taken into account is near 0. The relationship between them is purely due to their common association with the clusters we see (common cause scenario).

2.1. A second dataset

What about a similar situation where the relationship between predictor and outcome is not spurious? Consider the data in Figures 5 and 6.

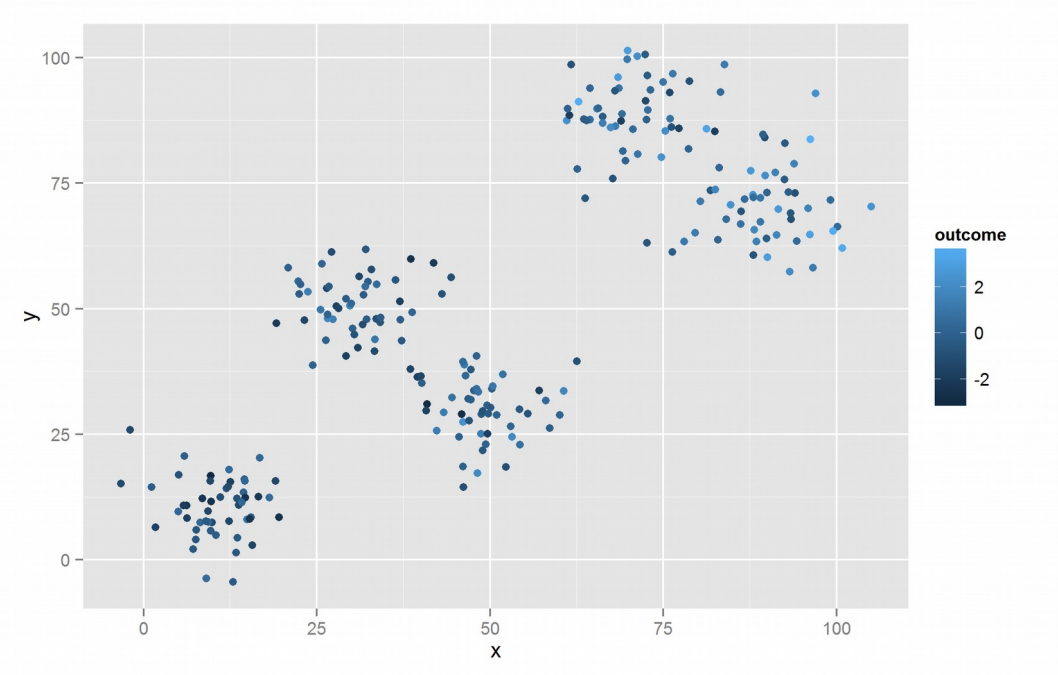

Figure 5: Dataset 2. Flatland and outcome.

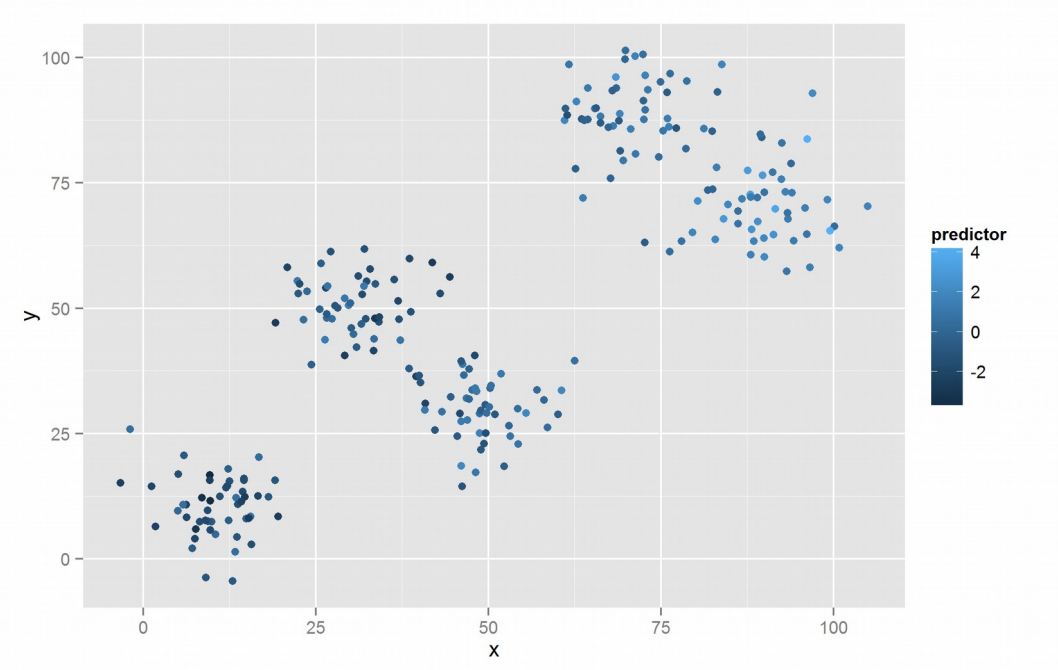

Figure 6: Dataset 2. Flatland and predictor.

We see the same overall pattern as before, but it is subtlety different. The correlation in the original dataset between predictor and outcome is nearly the same, .68 vs. .66 before. However, using the distance scores, the correlation between outcome and spatial distance is only .17 vs. .45 before. The distance-based correlation between predictor and outcome is .43 vs. .36 before.

The semi-partial correlation for predictor and outcome correcting the latter for spatial distance is .39 vs.

.13 before. A fairly large difference. Using multiple regression the betas are .42 and .04 for predictor and spatial distance, respectively.

Clearly, the approach of using distance scores shows that the association between predictor and outcome is different in the two datasets.

2.2. The relationship between correlations of distances and correlations in the original data

Before we go further, it is worth examining the relationship between correlations of distance data and correlations of the original data. From my experimentation correlations of distance scores seem to be an r^2-type statistic (Hunter & Schmidt, 2004, p. 189). This can be seen if we simulate normally distributed data with a known correlation and then correlate their distance scores. The results are shown in Figure 7.

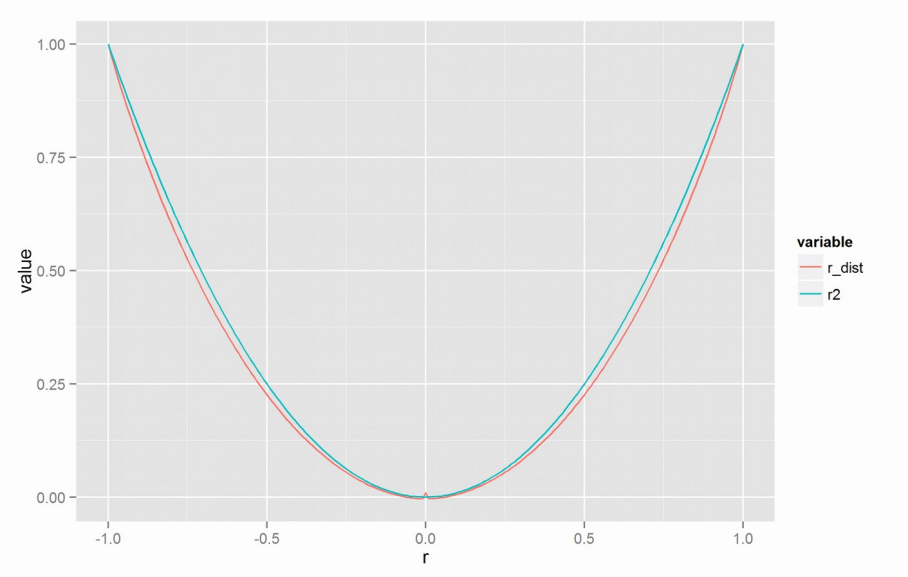

Figure 7: Comparison of r, r dist and r^2.

As can be seen, the correlation of distance scores is almost exactly the same as the square of the correlation, or conversely, the square root of the correlation of distance scores is almost the same as the original correlation score.

Going back to the results in the previous section, this means that we need to take the square root of the semi-partial correlations and standardized betas to get an r-type measure. These values are shown in Table 1.

Relationship	Dataset 1	Dataset 2
r (predictor x outcome), original data	.66	.68
sqrt. r (predictor x outcome), distance data	.60	.66
sqrt. r (spatial x outcome), distance data	.67	.41
sqrt. r_{sp} (predictor x outcome, spatial), distance data	.36	.62
sqrt. beta (predictor x outcome), distance data	.40	.65
sqrt. beta (spatial x outcome), distance data	.60	.20

Table 1: Table of numbers from datasets 1 and 2.

There are several things worth noting. 1) The square root conversion seems to work because the distance-based correlations are similar to the ones based on the original data (green cells). 2) distance-based correlations were higher for the first dataset as expected (dark green). 3) Semi-partial correlations of distance-data showed that the predictor had a substantial size (.36) in the first dataset despite this relationship being entirely spurious. However, it did correctly indicate a strong effect in the second dataset (.62) (orange). 4) Similarly, multiple regression showed that the predictor had substantial validity in the first dataset and reversely that spatial distance had some validity in the second dataset (light red). Neither should be the case. On the positive side, in both cases did the true causes have larger effect sizes (.60>.40; .65>.20).

3. k nearest spatial neighbors regression

Given the problems of the CD method, I developed a new method. It is a variant of the general approach of using the neighboring points to predict values, called k nearest neighbors (KNN) (James, Witten, Hastie, & Tibshirani, 2013). The implementations of KNN I could find for R did not support spatial data, so I wrote my own implementation which I dub *k nearest spatial neighbors regression* (KNSNR) The approach is fairly simple:

1. For each datapoint, calculate the distance (spherical or euclidean) to each every other datapoint.
2. For each datapoint, find the k nearest neighbors.
3. For each datapoint, calculate the mean value of the neighbor's scores and use that as the prediction.

One can alter k in (2) to tune the method. This value should probably be found thru cross-validation (James et al., 2013).

(3) can altered to make more advanced versions such as taking into account the relative distances among the nearest neighbors or the case weights (if using aggregated data or uncertain data).

The function I wrote will output one of three things: 1) predicted values, 2) correlation of actual values with predicted values, and 3) residuals. Each has its use.

The correlation of actual values with predicted values is a measure of the SAC in a given variable. It corresponds to the correlation of a given variable and spatial distance when using distance data. The values for the outcomes in examples 1 and 2 are .74 and .30 using k=3, .80 and .47 using k=50.

Which k is the right to use? All of them. What k does here is change the zoom level. The value at each k tells you how much SAC there is in the dataset given a particular level of zooming in on every case. Figure 8 shows the amount of SAC in the outcome variable according to the first 100 values of k for multiple datasets.

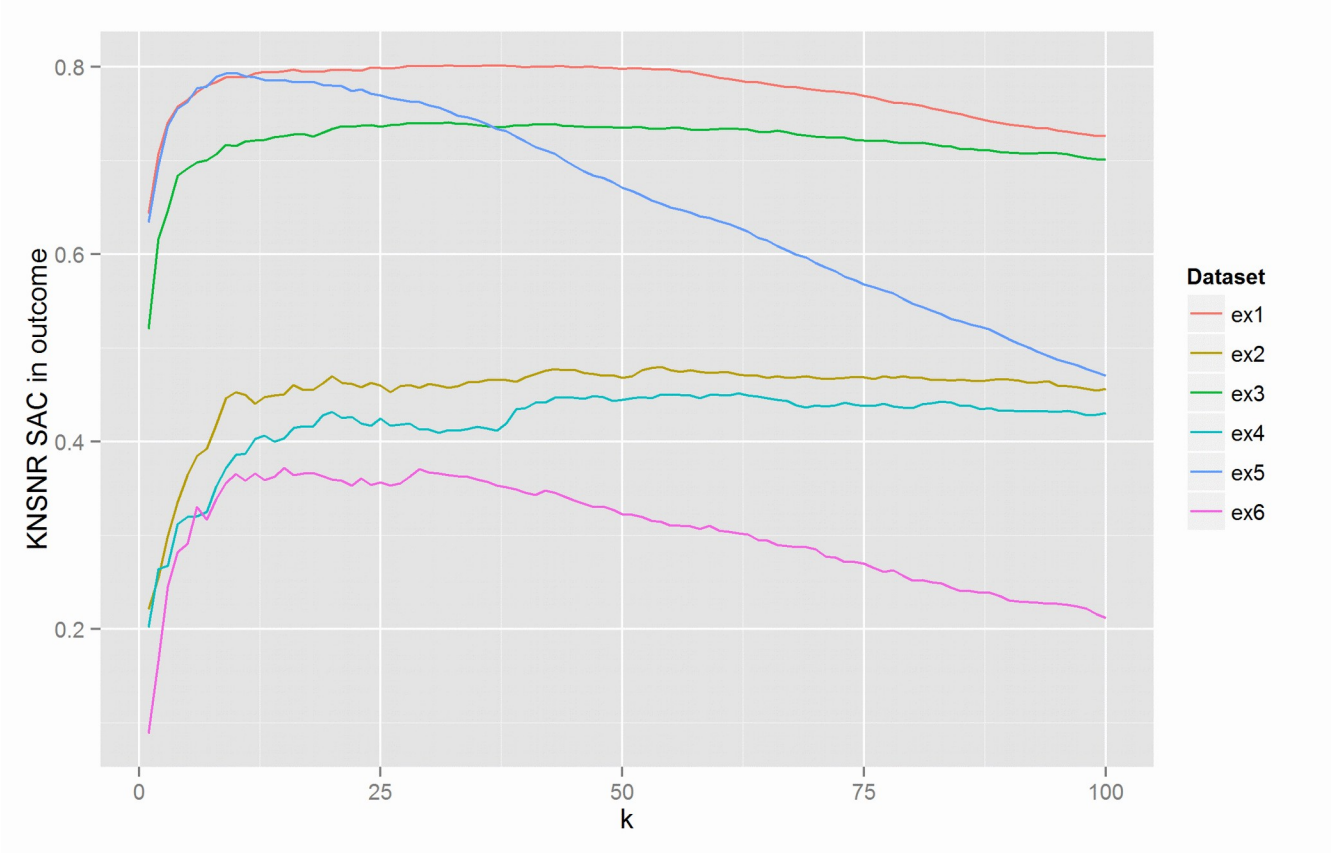

Figure 8: The amount of SAC in datasets 1-6 according to KNSNR at a k 1-100.

In general, k should be larger than 1. k=3 often works well. k reached plateaus for datasets 1-2 around k=10. For these datasets k=50 makes more sense because there are 50 cases in each cluster. At all levels, however, do we see that SAC is stronger for ex1, just as was found using the correlation of distance-values.

The predicted scores from KNSNR can be used in a multiple regression model to where they can compete with other variables. The residuals can correlated with a predictor to easily perform semi-partial correlation with SAC controlled for in the chosen variable. The semi-partial correlation between predictor and outcome controlling outcome for SAC in the first dataset is .01 while it is .42 in the second dataset (using k=50). In other words, the method tells us that the without SAC in the outcome the predictor doesn't work in the first dataset, but it does in the second. The semi-partial correlations of spatial prediction x outcome controlled for predictor are .34 and .01, for the first and second datasets respectively. In other words, spatial proximity retains predictive value once the predictor has been controlled, showing that there is an additional unmeasured cause with strong SAC. All four results are

good because they are in line with how the data were generated.

4. A different kind of spatial autocorrelation

The previous examples were easy in the sense that the data pretty clearly contained clusters of points. One could have easily classified the cases into the regional clusters and used a more traditional approach to the question: multi-level analysis. In simple terms, one would look whether relationship found in the pooled data holds when analyzing the data within each cluster. Table 2 shows these results.

Cluster	Dataset 1	Dataset 2
1	-0.176	0.511
2	-0.094	0.615
3	-0.037	0.5
4	0.117	0.511
5	0.162	0.617

Table 2: Correlations inside clusters in datasets 1 and 2.

Clearly, the datasets are markedly different in that there is a near-zero relationship inside the clusters in the first, but there are fairly strong relationships in the second.

When results are multiple levels give discrepant results it is termed *Simpson's paradix.* Not understanding this concept has lead to bogus claims of sex discrimination at least twice (CaAl, 2015; Kievit, Frankenhuis, Waldorp, & Borsboom, 2013). An interactive visualization of the phenomenon can be found at: http://emilkirkegaard.dk/understanding_statistics/?app=Simpson_paradox

Clearly we did not need to use SAC to spot the difference between datasets 1-2. However, the datasets are unusual in that they permit very easy clustering of the cases. Real life datasets are often more difficult to deal with. Consider datasets 3-4 shown in Figures 9-12.

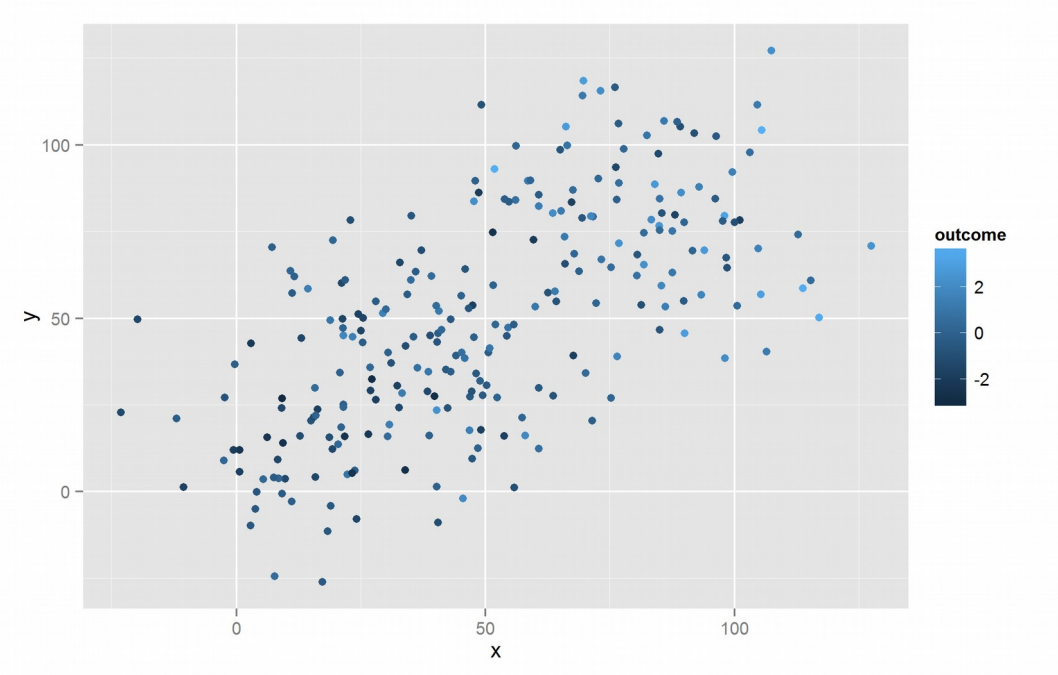

Figure 9: Dataset 3. Flatland and outcome.

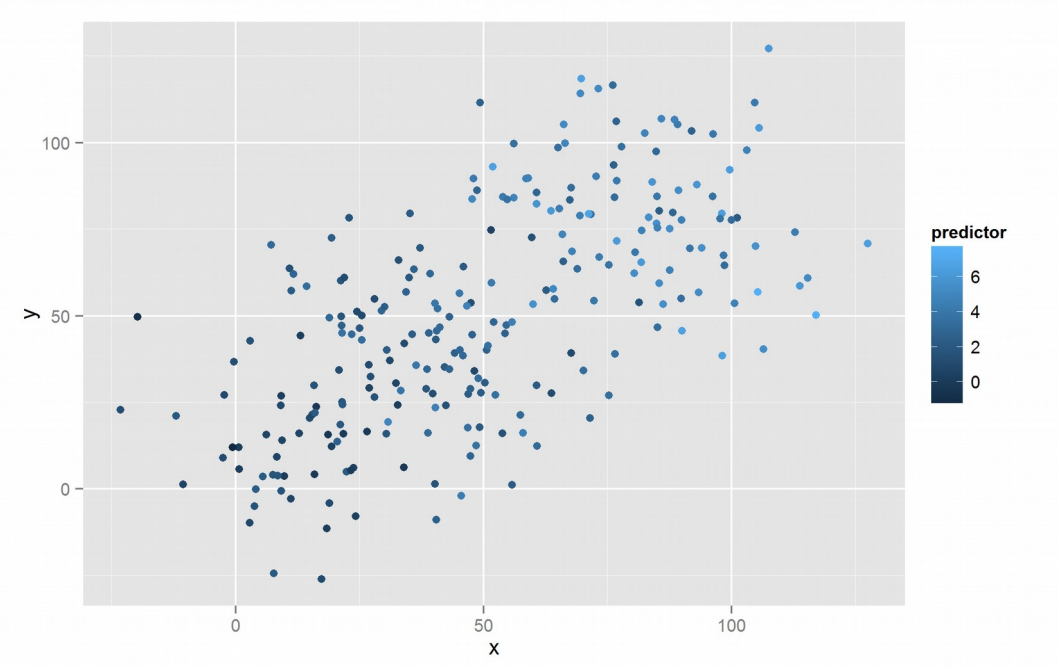

Figure 10: Dataset 3. Flatland and predictor.

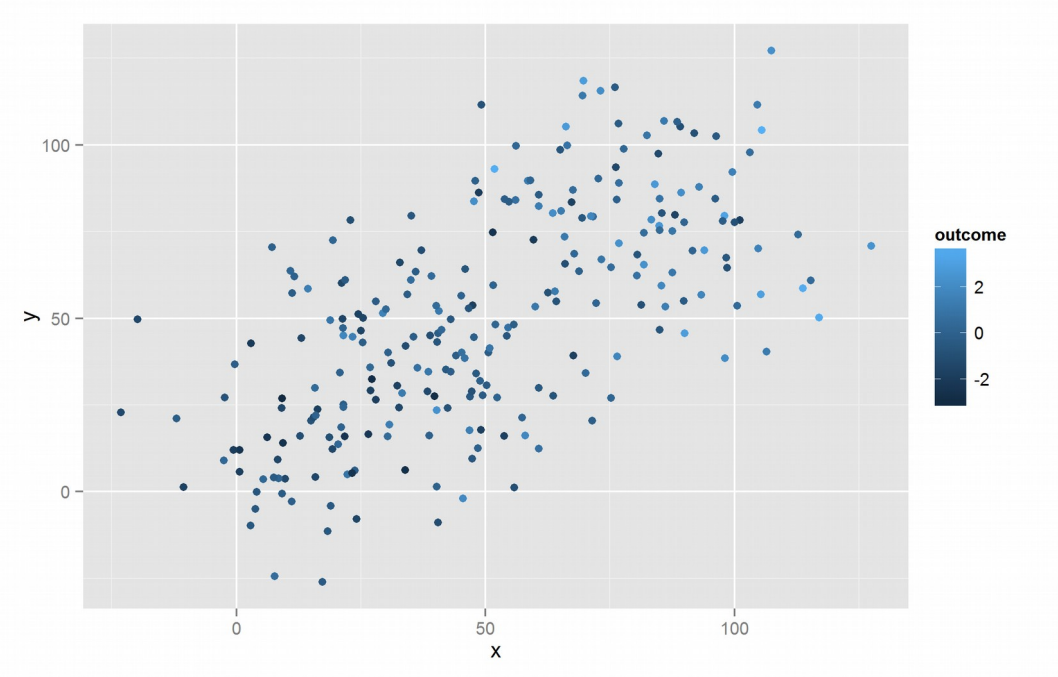

Figure 11: Dataset 4. Flatland and outcome.

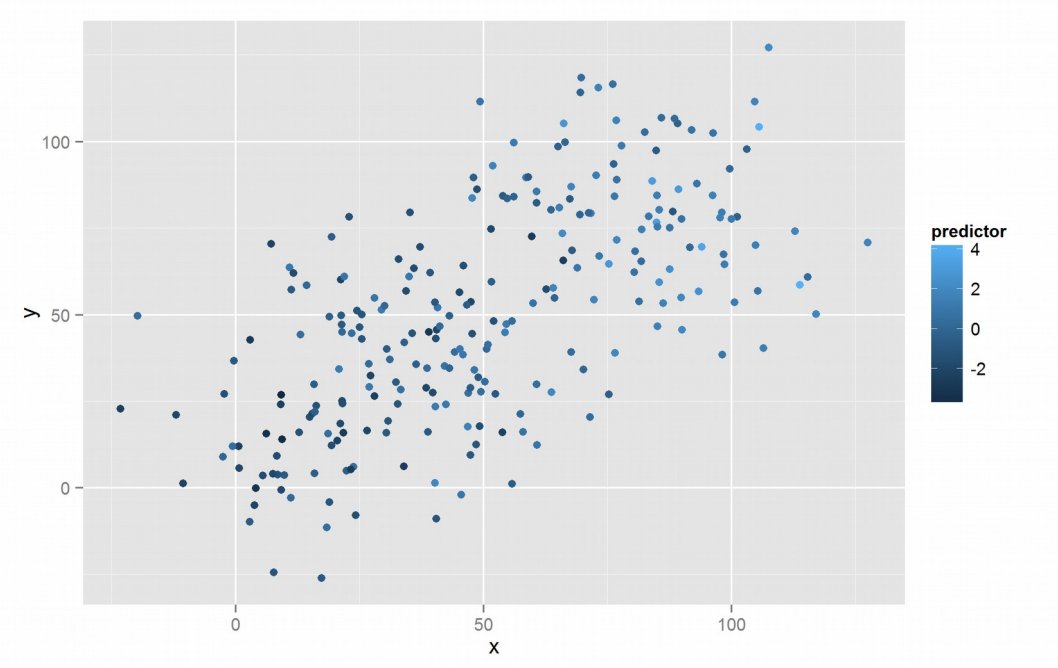

Figure 12: Dataset 4. Flatland and predictor.

These datasets are variations of datasets 1-2, yet now the clusters are not easy to spot. One would not be able to use a simple clustering algorithm on the spatial data to sort them into 5 clusters correctly. If one cannot do that, one cannot use the standard multi-level approach because it requires discrete levels.

The correlations between the predictors and outcomes is in fact identical to before: .66 and .68 respectively. These correlations are generated in the same way as before, meaning that the first is a spurious, SAC induced correlate, while the second is a true cause. The datasets still contain a sizable

amount of SAC as was seen in Figure 8, so one might still wonder how this affects the results. Using the semi-partial SAC control approach, the results are .15 and .46 respectively. Should the number .15 be closer to 0? Not exactly.

To understand why, I need to be more explicit about how the data were simulated. In the odd numbered datasets, the outcome and predictor variables are both a function of the clusters, which serve is a SAC variable (SACV). In the even numbered datasets, the predictor is a function of the clusters + noise and the outcome is a function of the predictor + noise. Figure 13 shows the path models.

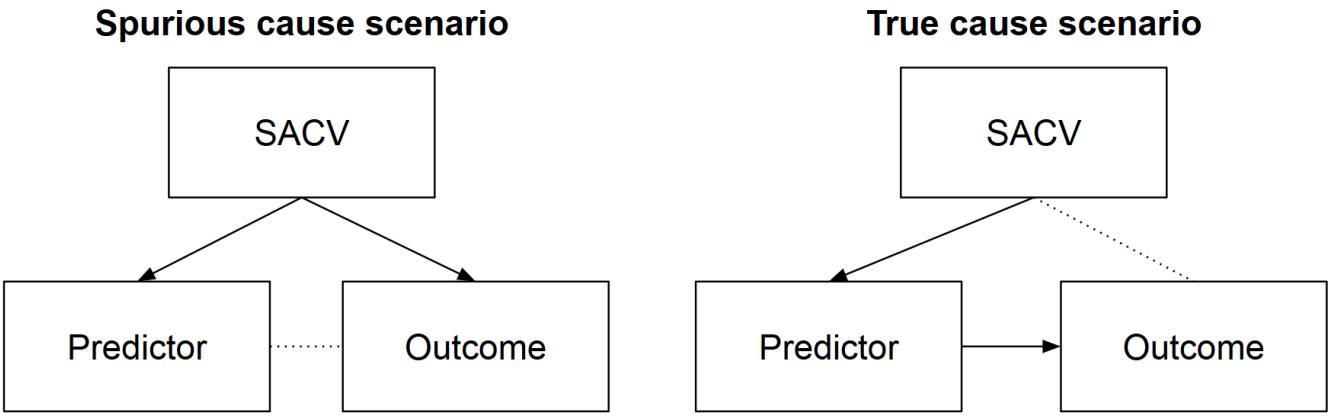

Figure 13: Path models for the example datasets.

The difference between examples 1-2 and 3-4 is solely that the within cluster spatial variation was increased from sd=6 to 15. This weakens the SAC in the data and thus the effect of controlling for SAC. The results are thus in line with expectations.

Still, we would like paired examples that appear similar -- i.e. has similar r (predictor x outcome) -- but where the first effect is spurious and the second is real and that our methods correctly identify this. To do this, I created two more datasets. This time there are no a priori clusters.

The generation proceeded like this:

1. For each datapoint:
    a) Sample two values from the uniform distribution of numbers between 1 and 100 and use these as x and y coordinates.
    b) Sample a number from a standard normal distribution and use it as the score for the SACV.
2. Induce SAC in the SACV using a KNSN-type approach.

After this, I proceeded similarly to before, namely that creating two variables, predictor and outcome, from according to the path models shown in Figure 13. Noise were added to hit a target uncorrected correlation of .66, similar to datasets 1-4. Because the SACV variable has strong SAC, controlling for SAC decreases the correlation in the spurious cause scenario because path goes thru the SACV, but not in the true cause scenario where causation is directly from predictor to outcome.

It is worth going into more detail about how SAC was induced. The algorithm works as follows:

1. For each iteration:

a) For each datapoint:

   1. Find the k nearest spatial neighbors.
   2. Find the mean value of the chosen variable for these neighbors.

b) For each datapoint:

   1. Change the value of the datapoint to (1-w) times its current value and w times the mean value of its neighbors.

(a) and (b) must proceed in this fashion otherwise the order of the cases would influence the algorithm. The algorithm requires values for the three parameters: i, k and w. i determines the number of iterations the algorithm goes thru. k control the number of neighbors taken into account and w controls the relative weight given to the value from the neighbors. Smaller k means the SAC pattern will be more local, while both i and w makes the SAC stronger but in different ways. The parameters used for datasets 5 and 6 were i=20, k=10, w=1/3.

Figures 14 to 17 show the spatial layout as well as predictor and outcome values for datasets 5-6.

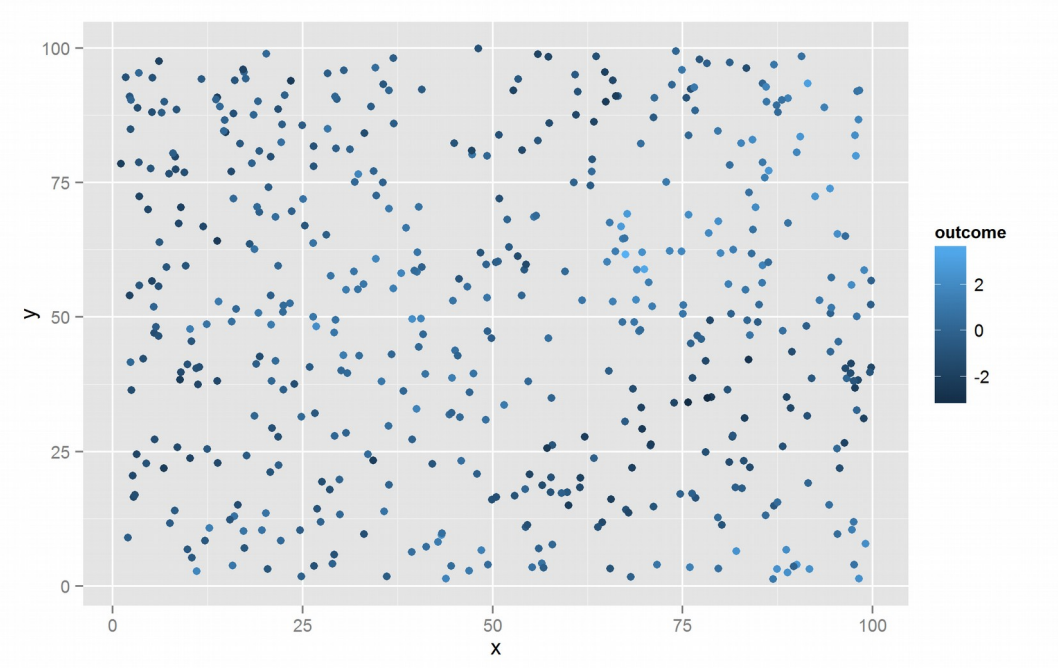

Figure 14: Example 5. Flatland and outcome.

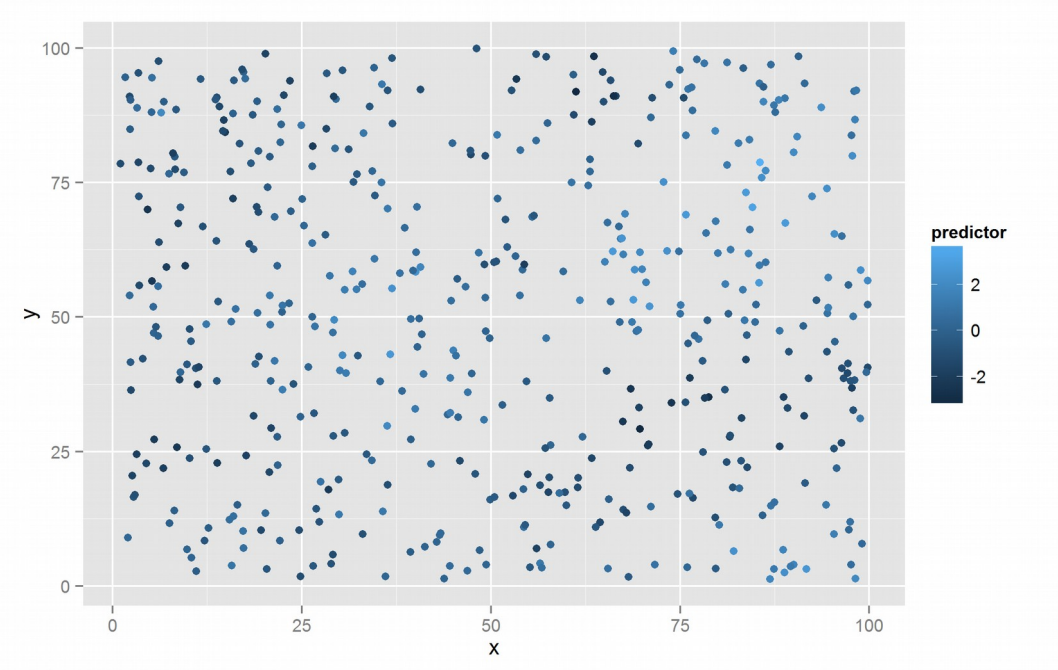

Figure 15: Example 5. Flatland and predictor.

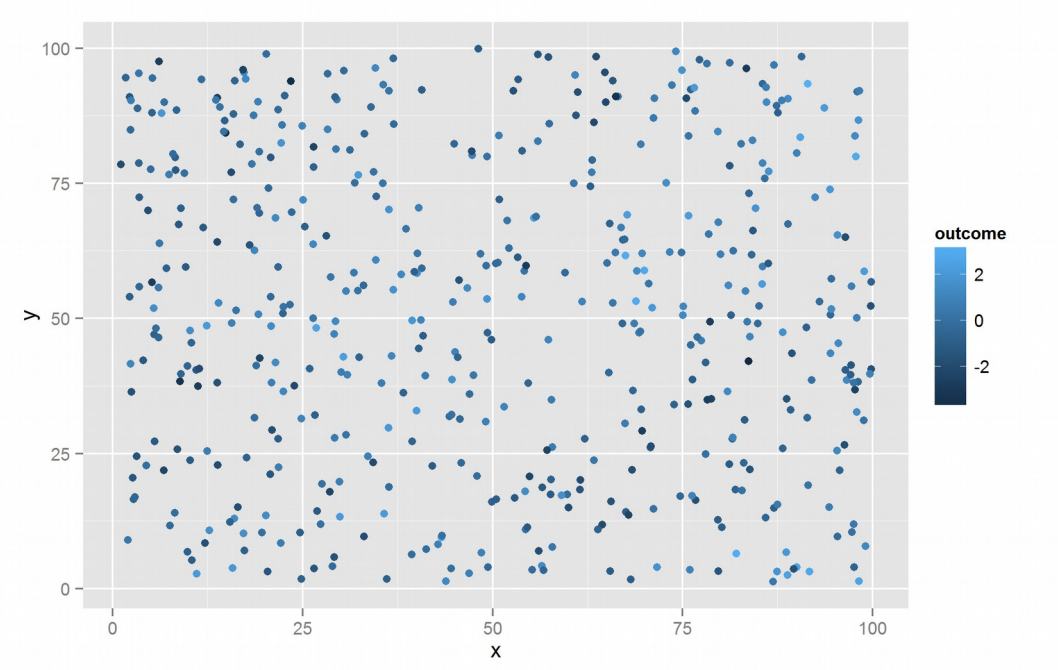

Figure 16: Example 6. Flatland and outcome.

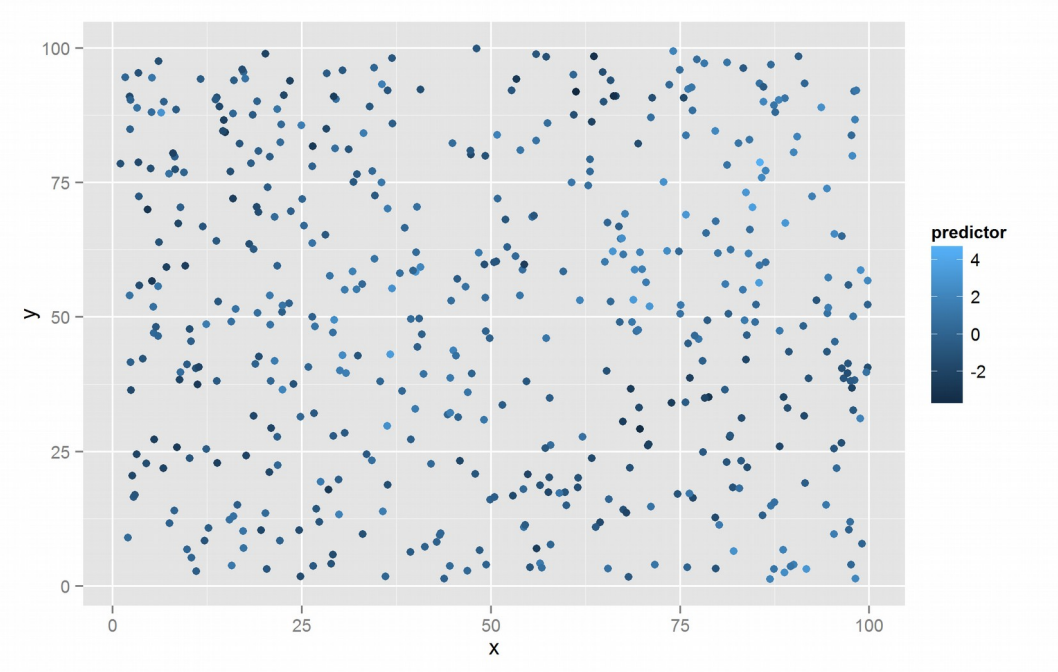

Figure 17: Example 6. Flatland and predictor.

From just inspecting the spatial plots, it is not easy to see a marked difference between datasets 5 and 6. Both datasets clearly show some degree of SAC. Figures 18 and 19 show the regressions between predictor and outcome.

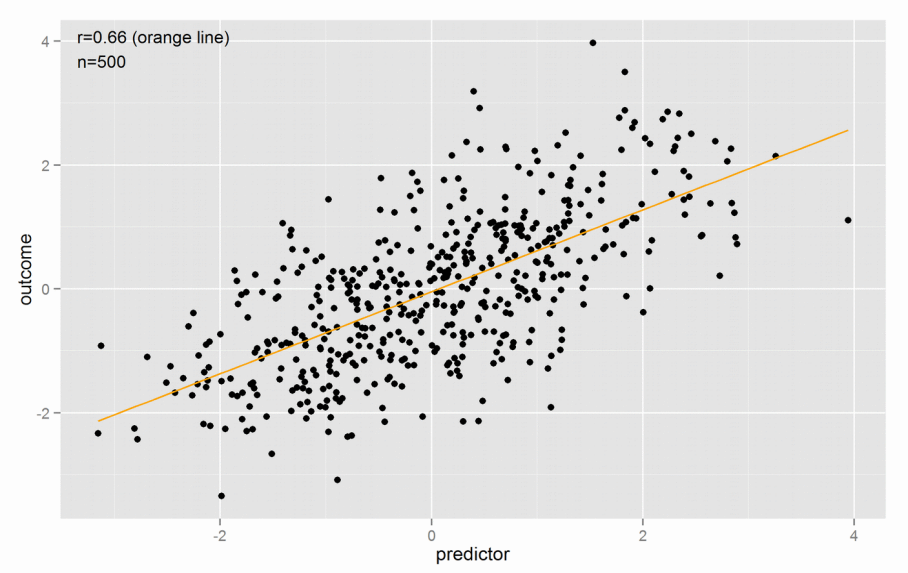

Figure 18: Scatter plot for predictor and outcome in dataset 5.

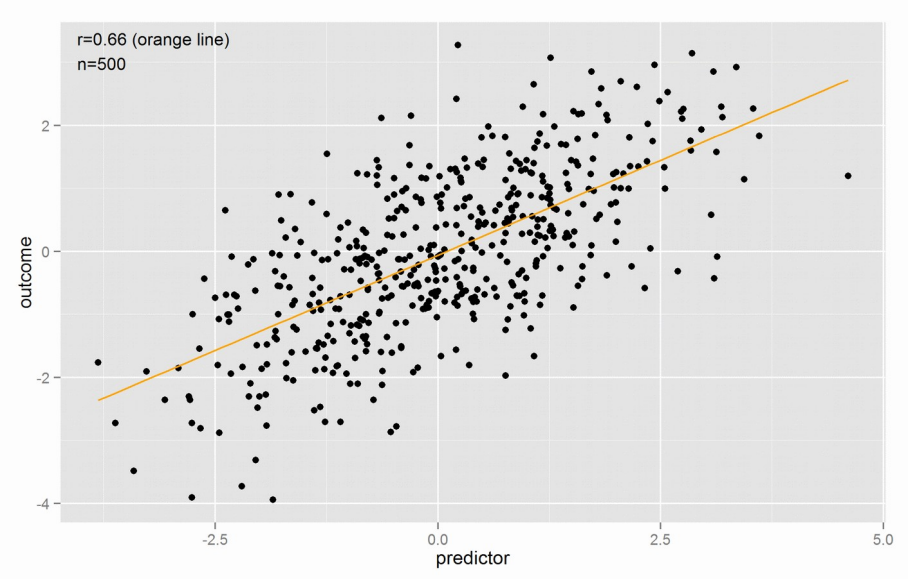

Figure 19: Scatter plot for predictor and outcome in dataset 6.

Altho not identical, there do not seem to be marked differences between the datasets. Thus, one could not tell the situations apart by simple inspection. There are also no obvious clusters one could classify cases into and use multi-level analysis.

However, if we examine the strength of SAC in the datasets, they are fairly different as shown in Table 3.

Variable	**Dataset 5**	**Dataset 6**
SACV	0.998	0.998
x	0.999	0.999
y	0.998	0.998
predictor	0.758	0.632
outcome	0.794	0.365
outcome_predictor_resids	0.28	-0.035

Table 3: SAC in datasets 5 and 6. SAC calculated using k=10 because data were generated using k=10.

Unsurprisingly, the coordinate variables and the SACV show extreme levels of SAC. The predictors diverge a bit in SAC, while the difference in SAC for the outcome variables is fairly large. This is as expected given the way the data were generated because in dataset 6, the outcome variable is 2 steps away from SACV, while it is only 1 step away in dataset 5. Also worth noting is the SAC in the model residuals in dataset 5, which was discussed by (Hassall & Sherratt, 2011) as indicating a problem. This indicates that one or more causes not included in the model are SAC.

The correlations with the predictor after controlling for SAC are .08 and .49 for datasets 5 and 6, respectively showing that the method can successfully detect which association is due to an uncontrolled common cause which is SAC.

5. Relationship to Moran's I

It is worth comparing the two measures of SAC examined in this paper to the more standard methods in the field. Two methods are widely used to measure the amount of SAC in a dataset: Moran's I and Geary's C (Gelade, 2008; Hassall & Sherratt, 2011; Radil, 2011). These two measures are described as approximately inversely related and often only the first is used. I also used only the first because I was unable to understand the implementations of the latter in R (e.g. in *spdep* package). Both Moran's I and Geary's C are global measures of SAC, altho local variations exist (Radil, 2011).

Because KNSNR tends to hit a ceiling quickly (e.g. in Table 16), the analysis here is focused on variables where this doesn't happen. For this purpose, the outcome, predictor and residuals from predicting the first with the latter are examined with all three methods in across all datasets. Results from different datasets with the 'same' variable were appended. Because KNSNR requires a tuning parameter, 3 values were chosen for this (3, 10, and 50). A square root version of CD was also added to see if the non-linearity inherent in the CD correlations would substantially change the relationship to the other measures.

To increase sample size and diversity of datasets, more datasets were created. Dataset 7 is a variant of dataset 6, but where SAC is also induced into the outcome and predictor variables. Dataset 8 uses the distance to the center as the basis for the predictor values and the outcome values are a noisy version of the predictor. Datasets 9 and 10 are variants of dataset 8 but where the distance to the points (75;75) and (100;100) are used, respectively. Dataset 11 had no SAC induced and functions as a null dataset. Dataset 12 has datapoints perfectly distributed in a grid pattern, perfect negative SAC for the predictor and uneven SAC for the outcome. The appendix contains scatter plots for these additional datasets. Table 4 shows the SAC values according to the 5 measures and Table 5 shows their intercorrelations.

Dataset and variable	Morans_I	cd	cd_sqrt	knsn_3	knsn_10	knsn_50
ex1_outcome	0.403	0.452	0.673	0.741	0.789	0.798
ex1_predictor	0.432	0.547	0.739	0.764	0.812	0.819
ex1_outcome_predictor_resids	0.074	0.062	0.249	0.227	0.283	0.325
ex2_outcome	0.146	0.172	0.415	0.3	0.453	0.468
ex2_predictor	0.298	0.309	0.556	0.598	0.671	0.685
ex2_outcome_predictor_resids	-0.002	0.02	0.141	-0.02	-0.021	-0.107
ex3_outcome	0.248	0.378	0.615	0.646	0.716	0.735
ex3_predictor	0.278	0.468	0.684	0.698	0.758	0.756
ex3_outcome_predictor_resids	0.038	0.054	0.233	0.16	0.271	0.303
ex4_outcome	0.09	0.176	0.419	0.268	0.386	0.445
ex4_predictor	0.18	0.263	0.512	0.507	0.601	0.635
ex4_outcome_predictor_resids	-0.004	0.04	0.199	-0.018	0.014	-0.059
ex5_outcome	0.136	0.066	0.258	0.737	0.794	0.671
ex5_predictor	0.14	0.055	0.234	0.693	0.758	0.662

ex5_outcome_predictor_resids	0.021	0.026	0.162	0.14	0.28	0.243
ex6_outcome	0.031	0.041	0.202	0.246	0.365	0.323
ex6_predictor	0.105	0.042	0.204	0.54	0.632	0.565
ex6_outcome_predictor_resids	-0.013	0.023	0.152	-0.092	-0.035	-0.101
ex7_outcome	0.176	0.065	0.256	0.98	0.994	0.729
ex7_predictor	0.219	0.101	0.317	0.987	0.997	0.831
ex7_outcome_predictor_resids	0.174	0.088	0.296	0.976	0.99	0.676
ex8_outcome	0.126	0.017	0.131	0.599	0.651	0.642
ex8_predictor	0.27	0.028	0.167	0.994	0.993	0.972
ex8_outcome_predictor_resids	-0.007	0.002	0.04	0.018	-0.01	-0.171
ex9_outcome	0.188	0.201	0.448	0.595	0.645	0.65
ex9_predictor	0.395	0.483	0.695	0.997	0.997	0.985
ex9_outcome_predictor_resids	-0.007	0		0.015	-0.02	-0.199
ex10_outcome	0.2	0.25	0.5	0.6	0.652	0.66
ex10_predictor	0.414	0.565	0.752	0.998	0.998	0.993
ex10_outcome_predictor_resids	-0.007	0	0.015	0.016	-0.016	-0.176
ex11_outcome	-0.002	-0.002		-0.027	-0.023	-0.04
ex11_predictor	-0.006	0.001	0.037	0.019	-0.004	-0.126
ex11_outcome_predictor_resids	-0.002	-0.002		-0.027	-0.023	-0.04
ex12_outcome	0.004	0.001	0.032	0.722	0.702	-0.609
ex12_predictor	-0.006	0		-1	0.978	-0.969
ex12_outcome_predictor_resids	0.004	0.001	0.032	0.722	0.702	-0.609

Table 4: SAC measures of tre variables in 11 datasets.

	Morans_I	**cd**	**cd_sqrt**	**knsn_3**	**knsn_10**	**knsn_50**
Morans_I		0.89	0.884	0.724	0.718	0.824
cd	0.882		0.968	0.501	0.493	0.639
cd_sqrt	0.876	1		0.487	0.533	0.736
knsn_3	0.822	0.652	0.523		0.689	0.761
knsn_10	0.767	0.591	0.587	0.879		0.547
knsn_50	0.931	0.831	0.809	0.793	0.721	

Table 5: Intercorrelations among 6 SAC measures in 11 datasets. Pearson correlations above diagonal. Spearman's rank-order below.[1]

1 The reader with a sense for detail might wonder how the rank-order correlations for cd and cd_sqrt differ. Taking the square root of a vector of numbers does not change their order, so how can the rank-order correlations be different? The explanation is that sometimes the square root is undefined because the original number was negative. The correlations

All correlations between the measures were sizable as one would expect. The CD measure showed non-linearity because the rank-order correlations were noticeably stronger.

Behind the strong intercorrelations lie some surprisingly inconsistencies. Consider ex1_outcome vs. ex7_outcome_predictor_resids. r_dist gives 0.452 and 0.088 while knsn_3 gives 0.741 and 0.976. What this seems to mean is that there is a strong local SAC (which knsn_3 measures) in the residuals but a weak global SAC (which r_dist measures).

The KNSN results for dataset 12 need special attention. Because the points are distributed in an exact grid pattern, many pairs of points are equidistant. This poses a problem for KNN-type approaches because they have to picked k nearest neighbors. In this case, the computer chooses one based on the position in the distance matrix which is an arbitrary choice. If one is examining data distributed in a perfect grid pattern, extra attention should be given to the choice of k. Notice also that cases close to the edges have no neighbors in that direction (this isn't Pacman world), so the algorithm must choose neighbors further away. This can give strange results.

6. Discussion and conclusion

k nearest spatial neighbor-type algorithms seem to be useful for analyzing and inducing SAC in datasets. Furthermore, because they generate predicted values, they can be used to control for SAC in a given variable in a dataset. This is unlike Moran's I which can only report SAC, but not try to correct for it (Gelade, 2008).

Correlation of distances (CD; (Piffer, 2015)) seems to be somewhat problematic. Like KNSN approaches it can be used to control for SAC. However, the simulated datasets analyzed here showed that controlling for SAC in that fashion is inadequate. It is worth examining whether limiting CD to the nearest k neighbors or neighbors within a radius of a given number of kilometers makes it a better measure. Another option is to weigh the datapoints by weights that diminish as the distance increases. All three of the suggested modifications attempt to deal with the problem that results from the globalness of the simple CD.

A second problem with CD-type approaches is that because distance scores are based on two datapoints, any measurement error in these scores will have a compound effect on the difference scores, but the matter is complicated (Trafimow, 2015). I'm not sure if this has an effect on the conversion back to an r-type measure as described in this paper, but it might.

A third problem with CD-type approaches is that the number of distances quickly become very large. E.g. for a dataset with 10,000 cases, the number of distance scores is 49,995,000. This makes for fairly resource-intensive calculations. Random sampling of datapoints could probably be used to offset this problem at the price of some sampling error.

6.1. Semi-partial correlation vs. multiple regression

In the paper I have mostly correlated the predictor with the residuals of the outcome variables once the SAC had been removed. This is a semi-partial correlation. Alternatively, one could use multiple

---

are based on pair-wise, not case-wise complete data, which results in a slightly different set of cases used for the correlations for cd and cd_sqrt.

regression (MR) and enter both the predictor scores from the SAC analysis against the predictor variable and let them compete. This approach sometimes works, but not always. When the outcome variable has strong SAC and the predictor has a strong correlation to the outcome, using MR results in multicollinearity and the associated problems. Therefore, if one does use the MR approach one should pay attention to multicollinearity in the models, e.g. by means of the variable inflation factor (Field, 2013; Gordon, 2015).

6.2. Limitations

Only 11 test datasets were examined. Testing the methods against each other on a larger and more diverse set of datasets might change conclusions somewhat.

Of the traditional measures of SAC, only Moran's I was used in the comparisons. Further research should compare the proposed measures with other already developed measures such as Geary's C, Gi, Gi* and local Moran's I (Radil, 2011, p. 19).

Another method used by (Hassall & Sherratt, 2011), spatial eigenvector mapping, to correct for SAC could not be used because I could not find a suitable R implementation.

Supplementary material and acknowledgments

Links to peer review thread, supplementary resources ...

The R source code for this paper is available at

Thanks to ...

References

CaAl. (2015). NWO, Gender bias and Simpson's paradox. Retrieved from http://blog.casperalbers.nl/science/nwo-gender-bias-and-simpsons-paradox/

Field, A. P. (2013). *Discovering statistics using IBM SPSS statistics* (4th edition). Los Angeles: Sage.

Gelade, G. A. (2008). The geography of IQ. *Intelligence*, *36*(6), 495–501. http://doi.org/10.1016/j.intell.2008.01.004

Gordon, R. A. (2015). *Regression analysis for the social sciences* (Second edition). New York: Routledge, Taylor & Francis Group.

Hassall, C., & Sherratt, T. N. (2011). Statistical inference and spatial patterns in correlates of IQ. *Intelligence*, *39*(5), 303–310. http://doi.org/10.1016/j.intell.2011.05.001

Hunter, J. E., & Schmidt, F. L. (2004). *Methods of meta-analysis correcting error and bias in research findings*. Thousand Oaks, Calif.: Sage. Retrieved from http://site.ebrary.com/id/10387875

James, G., Witten, D., Hastie, T., & Tibshirani, R. (Eds.). (2013). *An introduction to statistical*

learning: with applications in R. New York: Springer.

Kievit, R., Frankenhuis, W. E., Waldorp, L., & Borsboom, D. (2013). Simpson's paradox in psychological science: a practical guide. *Quantitative Psychology and Measurement*, *4*, 513. http://doi.org/10.3389/fpsyg.2013.00513

Piffer, D. (2015). A review of intelligence GWAS hits: Their relationship to country IQ and the issue of spatial autocorrelation. *Intelligence*, *53*, 43–50. http://doi.org/10.1016/j.intell.2015.08.008

Radil, S. M. (2011). *Spatializing social networks: making space for theory in spatial analysis*. University of Illinois at Urbana-Champaign. Retrieved from https://www.ideals.illinois.edu/handle/2142/26222

Trafimow, D. (2015). A defense against the alleged unreliability of difference scores. *Cogent Mathematics*, *2*(1), 1064626. http://doi.org/10.1080/23311835.2015.1064626

Appendix

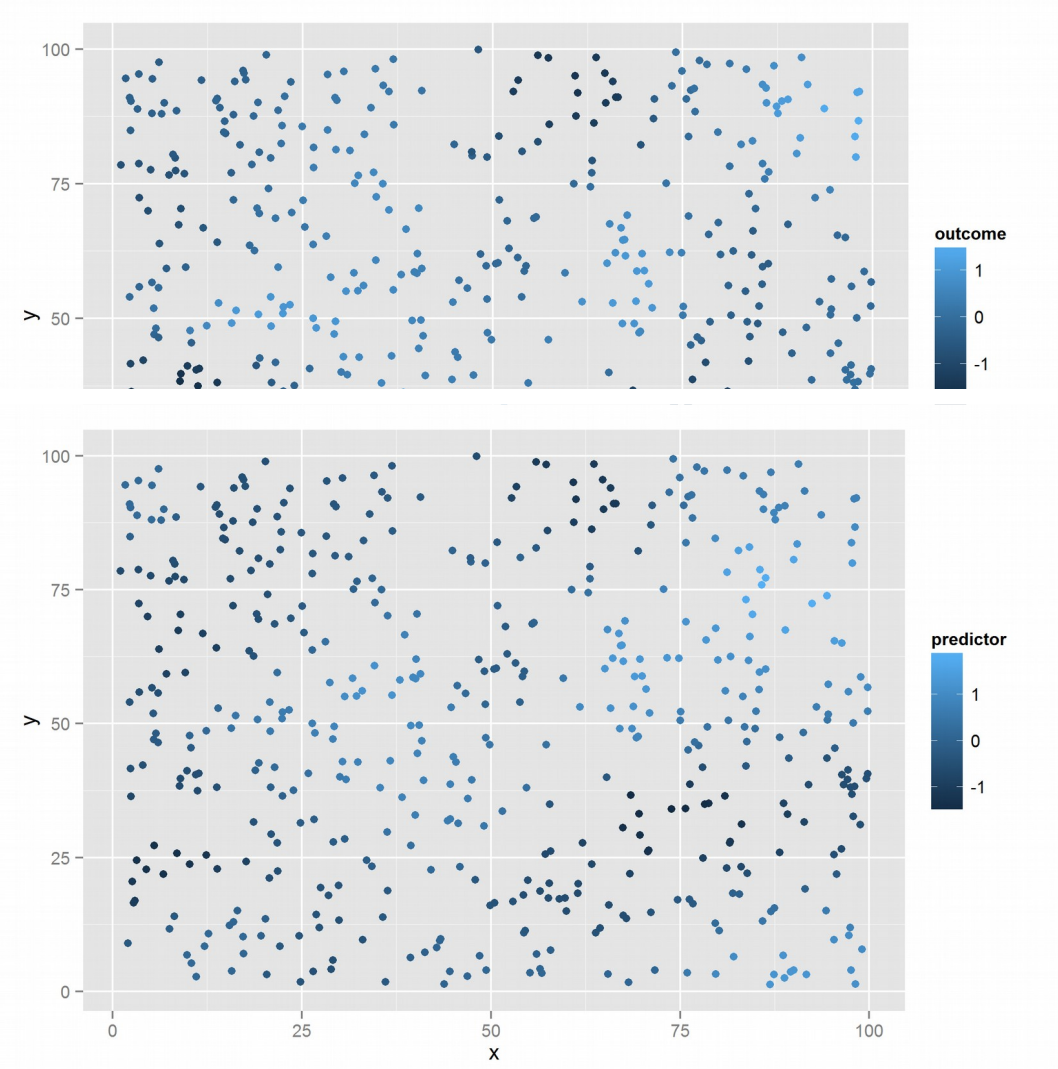

Figure 21: Dataset 7. Flatland and predictor.

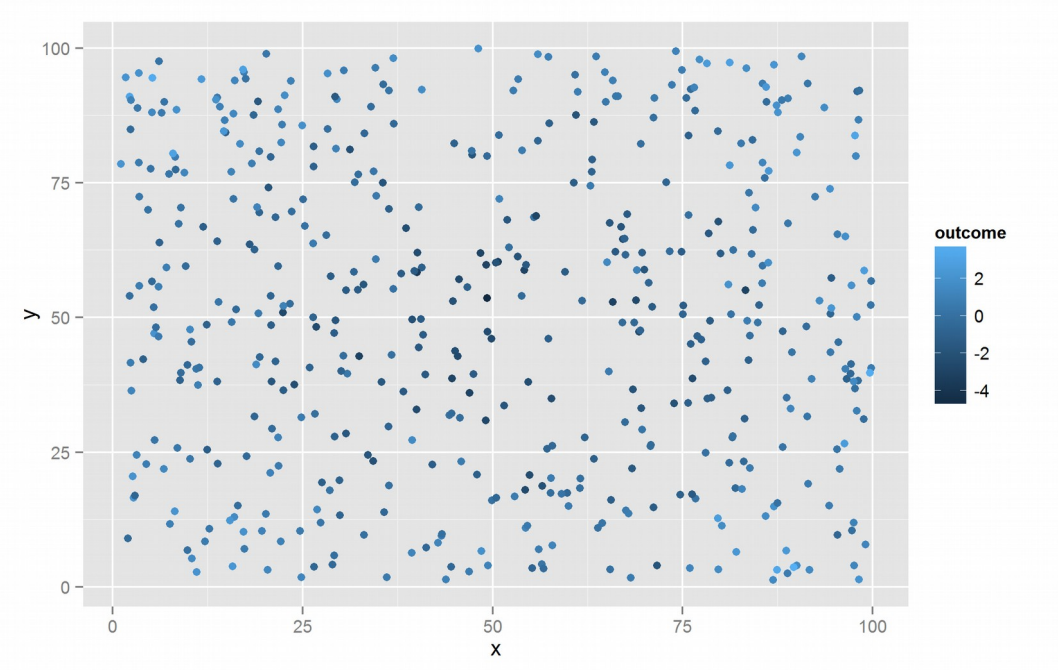

Figure 22: Dataset 8. Flatland and outcome.

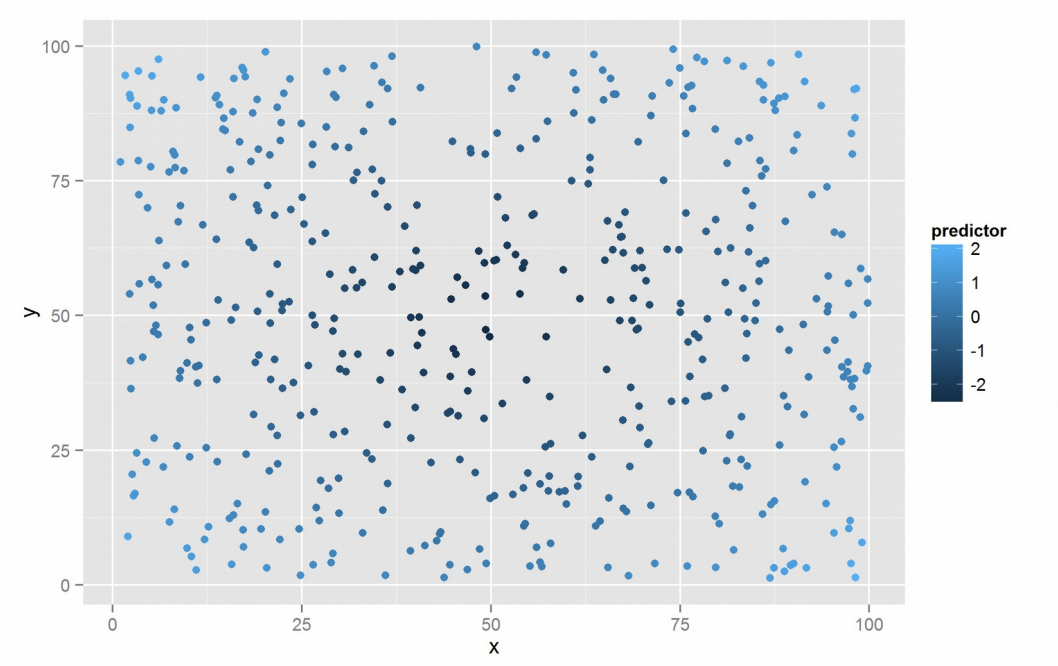

Figure 23: Dataset 8. Flatland and predictor.

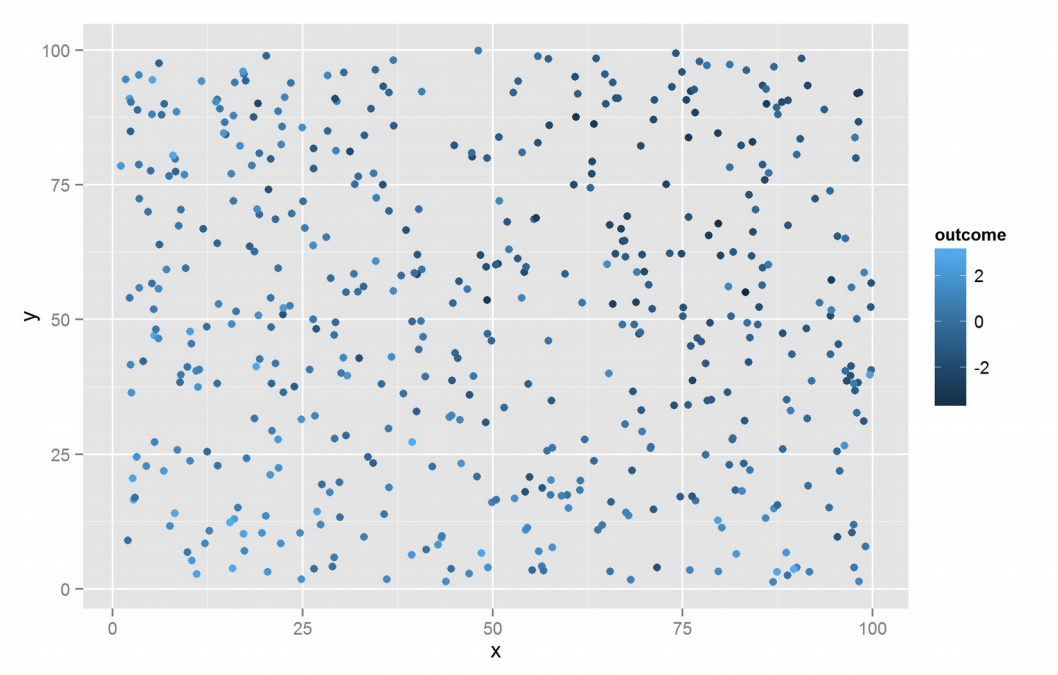

Figure 24: Dataset 9. Flatland and outcome.

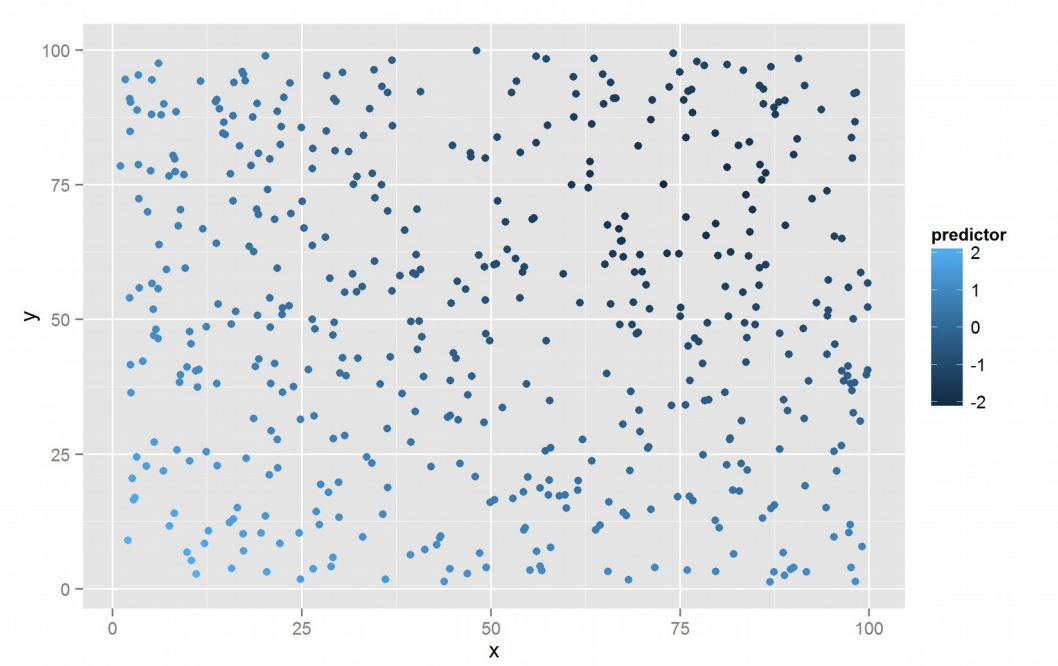

Figure 25: Dataset 9. Flatland and predictor.

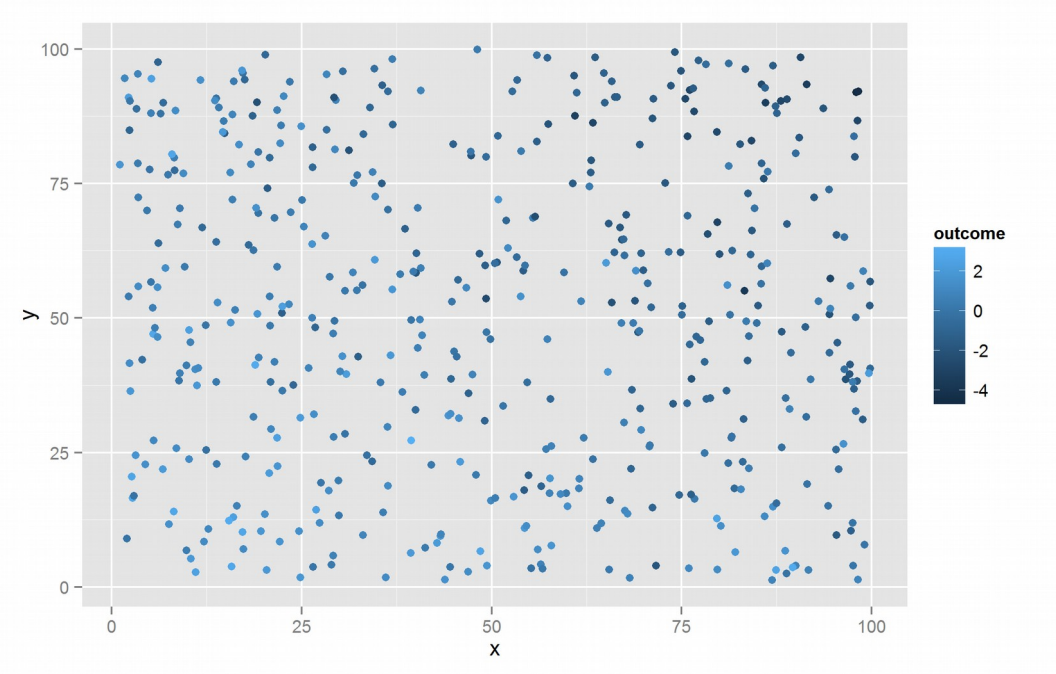

Figure 26: Dataset 10. Flatland and outcome.

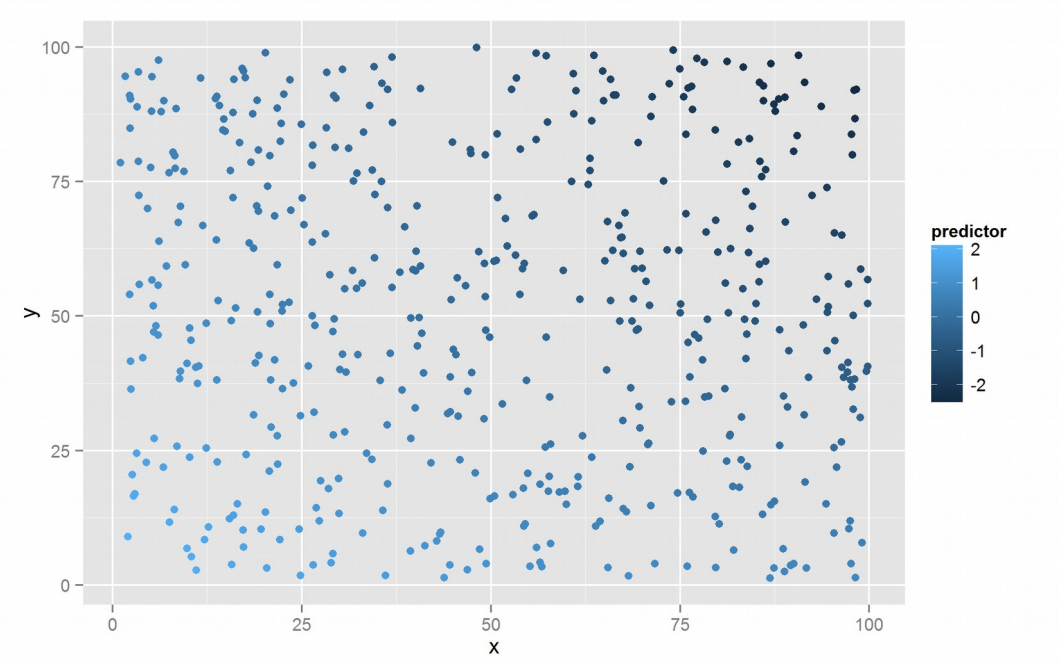

Figure 27: Dataset 10. Flatland and predictor.

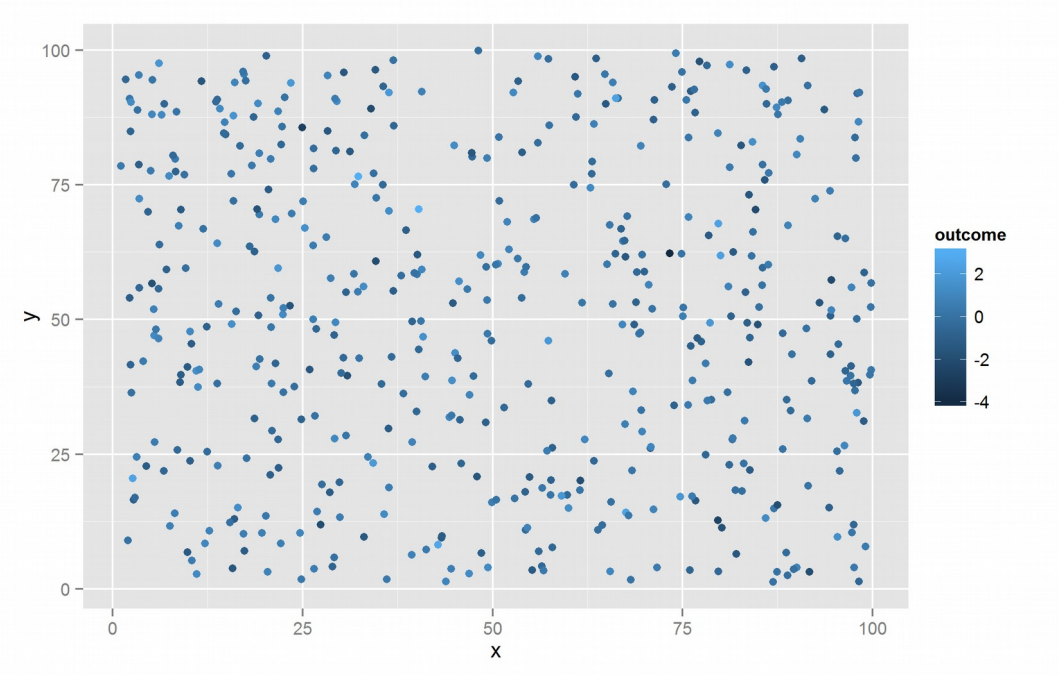

Figure 28: Dataset 11. Flatland and outcome.

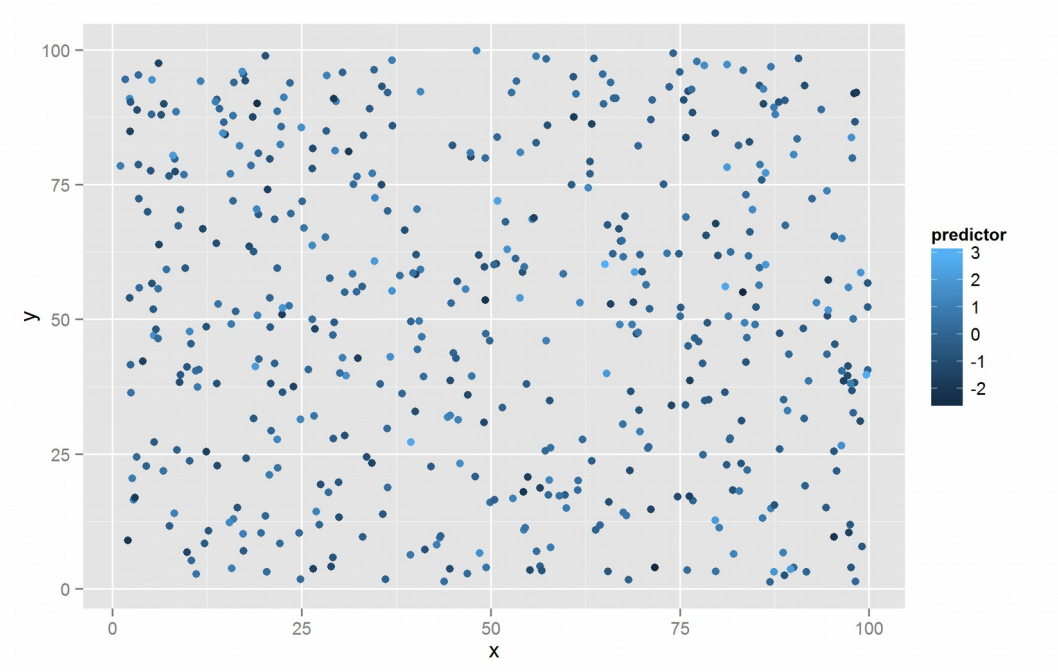

Figure 29: Dataset 11. Flatland and predictor.

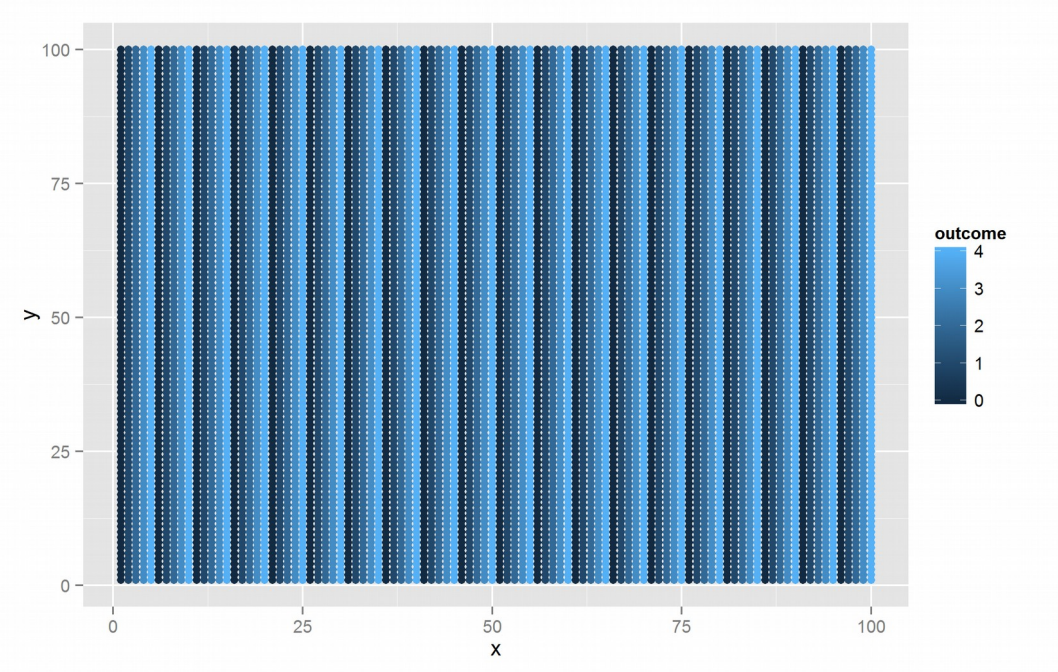

Figure 30: Dataset 12. Flatland and outcome.

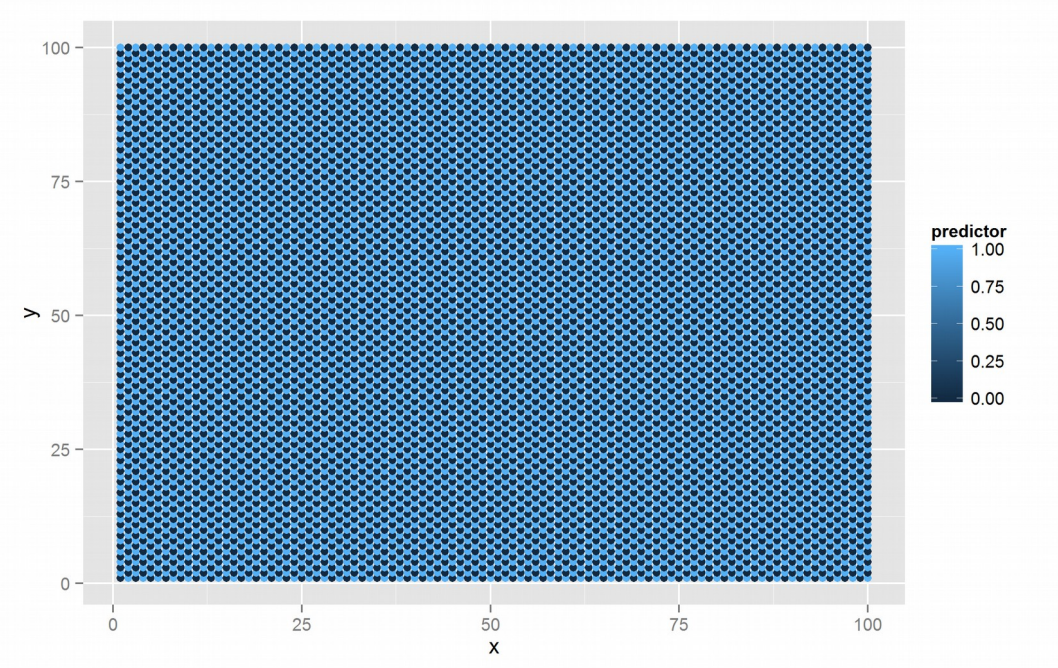

Figure 31: Dataset 12. Flatland and predictor.