Digital Journalism

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/rdij20

# Should I Stay or Should I Code? Of Collaboration and Do-It-Yourself Programming in Investigative Journalism

Espen Sørmo Strømme

**To cite this article:** Espen Sørmo Strømme (17 Oct 2023): Should I Stay or Should I Code? Of Collaboration and Do-It-Yourself Programming in Investigative Journalism, Digital Journalism, DOI: 10.1080/21670811.2023.2261030

**To link to this article:** https://doi.org/10.1080/21670811.2023.2261030

DIGITAL JOURNALISM
https://doi.org/10.1080/21670811.2023.2261030

RESEARCH ARTICLE

OPEN ACCESS

# Should I Stay or Should I Code? Of Collaboration and Do-It-Yourself Programming in Investigative Journalism

Espen Sørmo Strømme

NLA Hogskolen Bergen - Kristiansand, Journalism and Communication, Kristiansand, Norway

**ABSTRACT**
For journalists, numerous digital tools such as spreadsheets or web application are readily available for easy implementation. However, using programming to build or modify scripts has been an unexplored option for many journalists. Programming requires skills from the ICT professions, so when journalists find coding to be an efficient tool, do they stick with their old methods and ask ICT specialists for help, or do they learn to code themselves? This article is a qualitative study of the methodological reports pertaining to investigative projects nominated for the Norwegian journalism award SKUP. In these reports, programming has been used as a tool in 109 projects. The study examines which problems are solved using coding and the arguments for choosing this tool over simpler methods. It also looks at who performs the coding and how the editorial teams collaborate to reap the benefits of coding. The findings show that many journalists often find it difficult to collaborate with ICT specialists. They would therefore rather learn basic coding skills themselves. They follow a "do-it-yourself" philosophy and learn simple and open programming languages such as SQL, Python just well enough to achieve their goals.

**KEYWORDS**
Investigative journalism; digital journalism; programming; cross-disciplinary collaboration

## Introduction

This study examines the reasons for investigative journalists to use programming as a tool, and how they implement it. I examine the journalist's own descriptions and choices of methods and implementations through the large material of SKUP reports written since 1991. A computer program is a "detailed plan or procedure for solving a problem with a computer" (Britannica, T. Editors of Encyclopedia, 2022). A computer program may also be referred to as "an algorithm" (Diakopoulos 2015; Zamith and Haim 2020). Digital tools in general are known to be useful in all parts of the journalistic work process: to harvest, abstract, sort, interpret and visualise data (Parasie and Dagiral 2013, Bounegru, 2012) These tools are often multipurpose, like spreadsheets for analysing data or tools for creating digital maps. They require few skills and knowledge to use. Using code to build algorithms is a more tailor sewed method

**CONTACT** Espen Sørmo Strømme ✉ espen.stromme@nla.no

to solve a specific problem, and it requires the users to master skills of computer science. Both learning to code and implementing it is time consuming. One may expect this only to be used in special occasions and by collaborating with other professionals, described as "intraloper" or even external expertise, like hackers (Boyles 2020; Lewis and Usher 2014).

There are only a few studies that scrutinise the cases where journalists use programming as a tool in the making of news. Previous studies have examined the creation of news apps for mobile phones (Ananny and Crawford 2015), and how algorithms personalise content (M. S. Weber and Kosterich 2018). Other studies place more emphasis on the finished news publications rather than the journalistic process (Schiffrin, Tannen, and Hamilton 2015, 416–417). However, computer journalism encompasses the whole production process and there is a gap in knowledge of the earlier stages of the process, as also mentioned by several scholars (Schiffrin, Tannen, and Hamilton 2015; Wahl-Jorgensen and Hanitzsch 2019, s. 416–417).

Several studies map the need for learning digital tools in general, and what obstacles are in the way of acquiring such skills (Appelgren and Nygren 2014; Bowers, Rogers, and Schwabish 2017; Heravi 2018; W. Weber, Engebretsen, and Kennedy 2018). This is also reflected in studies about award-winning data journalism (Adegboyega and Heravi 2017; Young, Hermida, and Fulda 2018) although these mostly scrutinise the presentation and storytelling elements.

At the same time, there is a debate among educators whether coding should be part of the curriculum of journalism schools (Foust and Bradshaw 2020). Some fear that this would draw attention away from basic skills such as interviewing and reporting (Khazan 2013). Bowers et al. show that journalists question the return of investment when it comes to learning to code. They also find it difficult to decide what tools and languages to learn with respect to what will give the greatest value for their efforts (Bowers, Rogers, and Schwabish 2017). Splendore et al. show that it is much more common for European journalists to purchase help from external experts when they need programming than to learn to code themselves (Splendore et al. 2016). According to Stavelin, journalists who work in teams with developers perceive the developer's efforts as "something external," i.e., beyond the boundaries of journalism. In his words: "The blend of technical skill and journalistic values is a very rare mix. A more traditional technologist would need to muffle his (or her) enthusiasm for technology and align to the cultural climate in the newsroom." (Stavelin 2014).

How much coding journalists need to learn is also debated (Raja and Jun 2014), and eventually, this discussion in turn leads to further debate on the demarcation of journalism boundaries (Carlson and Lewis 2015; Gieryn 1983; Wiik 2015). This shows that further research about how programming may be used, and whether the journalists should learn this themselves or rather collaborate with experts is important, both for educators and practitioners.

**My research question is thus:**

RQ: How does investigative journalists use coding in their work process?

This leads to the following two sub-questions:

*SQ1: What are the arguments for choosing coding?*

*SQ2: In what cases do the journalists learn coding themselves, and when do they collaborate with ICT specialists?*

Furthermore, I look at whether programming is appreciated by the SKUP jury, through awards and diplomas of recognition, regarding coding as a useful new journalistic method, which impose the last sub-question:

*SQ3: How is programming valued in the field of journalism?*

## The case of Norway and the SKUP Award

From its establishment in 1991, the idea of the SKUP organisation was not just to establish an award for investigative journalism, but also to facilitate sharing and collaboration amongst journalistic peers across newsrooms, inspired by the Investigative Reporters and Editors (IRE) organisation (Ottosen 1996; SKUP 2010). The SKUP Award stands out from its international counterparts because it has a 30-year portfolio of methodology reports written by the journalists themselves. In these reports, the journalists explains how they have carried out their work, which obstacles they met and how they solved them. In one of the SKUP reports, the journalist expresses his understanding of the term "journalistic method" as: "Here, I will explain the method we used so that anyone may do exactly the same as we did." Contributing to developing new journalistic methods is one of the weighted criteria for winning the SKUP Award (Olufsen 2019).

These descriptions may sometimes be rich and valuable, other times superficial. Earlier studies show that this empirical source is especially valuable regarding the initial stages of the journalistic workflow. The journalists writing the reports emphasise the descriptions of collecting, cleansing, and analysing data while placing less focus on storytelling and presentation (Strømme 2020, 2022).

In recent years, these reports have been up to 20 pages long each. The complete material in this study consists of 1,458 reports, counting some 18,000 pages, whereof 109 of the works contain descriptions of coding.

The methodological reports are created for a different context, as they are intended to provide documentation and grounds for winning an award. This entails both advantages and disadvantages. The journalists make the selections themselves, and there is no threshold for nominating a work. Furthermore, the jury members are experts in the field of journalism and the winners will thereby constitute and reinforce the values in their field, giving the members an autodidactic representation of what journalists promote as "good" or" exceptional" etc. (Loosen 2021). In the case of the SKUP reports, the selection consists of detailed descriptions of the journalists' investigations, making it possible for the researcher to skip the work of selecting and sampling the data and instead focus on the analysis. This makes it possible to evaluate a vast number of works, or even all of them.

Since the SKUP Award is for investigative journalism in general, only a minority of the projects contain descriptions of coding. Nevertheless, when coding is chosen as a method, the journalists are expected to argue why this has been done and what kind of problems they solved in this way. The reports can also include descriptions of trial-and-error processes, for example tasks the journalist consider suitable for coding, but that was proved not to be.

The SKUP reports are clear expressions of how journalists compete to construct, reiterate, and challenge the boundaries of accepted practices, as described by Carson and Farhall (2018). Thus, introducing coding, or new ways of using code, is a way to challenge and possibly expand the field. Precisely what these journalistic practices are and how they are being developed and negotiated among practitioners is, however, a topic that appear to be far less covered in the literature (Bjerknes 2020; Gynnild 2007; Krumsvik et al. 2019).

## Research Design and Theory

### The Arguments for Programming

Flew at al. (2012) refer to Klein, Moon, and Hoffman (2006) to describe sense-making in computer science to construct a list of six arguments for using computational tools in the journalistic work process. These are: (1) fuse large amounts of data into succinct meanings, (2) process meanings in contextually relative ways, (3) Enable humans to achieve insights from this data fusion and processing, (4) infer hypotheses that humans are considering, (5) enable people to have access to intuitions of others, and (6) present information in relevant ways that enhance the tacit knowledge of humas about the subject.

Several scholars point out that the reasons for using computational tools are to automate, by building algorithms and abstract information. Automation is used to gather, sort, and interpret (Parasie and Dagiral 2013), and is typically performed by software like Excel. Spreadsheets are well-suited for solving commonly known problems of a certain scope. However, if the size of the data exceeds the limits or the sorting of the data becomes too sophisticated, journalists may identify the need to use databases, possibly with the programming language SQL or similar, or algorithms built by code. Thus, programming may be used to solve tasks that would otherwise be impossible due to the size of the data.

While repeated manual punching and cleansing of data will result in some degree of human errors, using scripts does not. Errors in the scripts are easier to fix and the altered code may quickly be executed again. This is not the case for manual cleansing, for example when using multiple Excel sheets. Cleansing these manually for a second time is also time consuming. Possible errors are also better documented with scripts because the code itself is testable, giving transparency to the whole process.

At its highest level, computational journalism may be used to abstract, or understand datasets, which includes algorithms to provide suggestions for sense-making and new ideas, for example through machine learning or advanced statistics. Digital tools may also be used to enable more accurate cross-referencing, data matching and pattern identification, both for abstraction and fact checking (Flew et al. 2012).

In addition, programming is described used as a way to collaborate with the audience and hackers. (Boyles 2020; Lewis and Usher 2014).

By using the list created by Flew et al. along with these other expectations, arguments for using coding are expected to be divided between the following categories:

1. Automation
2. Tasks that would otherwise be impossible to do
3. Fact checking and quality control of data
4. Abstractions: sense-making and new ideas
5. Visualisation
6. Crowdsourcing or other ways of including interlopers

When looking at the specific programming languages in use, Coddington shows that journalists are drawn to open-source software (2015), enforcing principles of transparency, iteration, tinkering and participation (Lewis and Usher 2013). The languages mentioned are R and Python (Simon 2021), which are both free and easily accessible. It is also common to find sharing and reuse of code, for example *via* Github.

### *To Learn or to Collaborate*

When the editorial teams have determined their reasons for coding and which language to use, they must subsequently implement it. If the team have no prior experience with coding, they have to decide whether they should learn to code themselves, collaborate with ICT specialists or a combination. Data journalists describe themselves as working within an interdisciplinary community that cross the traditional boundaries of journalism (Appelgren and Nygren 2014; Haim 2022; Lewis and Usher 2013). There are a range of proposals about how journalists can include knowledge and skills from other professions, spanning from including ICT specialists in the team, using external and internal interlopers, so-called “intralopers” (Holton and Belair-Gagnon 2018) or by crowdsourcing. Hackathons and meetups have been arranged by several occasions (Boyles 2020), for instance by organisations such as “Hacks/Hackers,” since before 2010 (Lewis and Usher 2014). Stencel and Perry (2016) query that aimed to identify skills that are highly ranked by editorial employers reveals that two-thirds of the organisations considered “coding/development” as essential. They identified two categories of coders: code-friendly journalists, journalists who have learned basic coding; and editorial-friendly coders, ICT specialists with enough understanding of news production to be able to engage in collaboration with journalists.

### *Routines and Bureaucracy Versus Improvisation and Ad-Hocracy*

Journalistic production at large has been described as a routine (Deuze 2005). Shoemaker and Reese (1996) describe this as “those patterned, routinized, repeated practices and forms that media workers use to do their jobs.” Thus, journalists perform similar or the same activities to produce the daily news. News stories are pipelined through several “beats,” undergoing predictable processes for gathering, analysing, processing, and presenting news (Tuchman 1978). These activities follow a professional logic (Lewis 2012) and a distinct occupational ideology (Deuze 2005), typically characterised as a professional group (Abbott 2014).

Such a depiction resemble what Weber (1958) describes as "bureaucracy." The bureaucracy is characterised by having articulated objectives, sharp divisions of labour, clearly defined tasks, well-developed hierarchies, and formalised systems of control. When several professions work closely together, the term "professional bureaucracy" can be applied. Achieving coordination therefor relies on the standardisation of skills and associated design parameters, training, and indoctrination (Mintzberg 1979). The benefit of such rule is efficiency, but it is unsuited for tasks that demand creativity and problem solving.

For knowledge-intensive firms, the workflow may be much more dynamic, demanding ad hoc solutions and strong decision-making (Kärreman, Sveningsson, and Alvesson 2002). The counterpart of a bureaucracy is "adhocracy," which Mintzberg (1979) describe as "… one that is able to fuse experts drawn from different disciplines into smoothly functioning ad hoc project teams."

In investigative journalism, projects become more knowledge-intensive and complex than in daily news production, and Tuchman's descriptions of routinised work may not be appropriate. The setup of teams in investigative journalism could be compared to Mintzberg & McHugh's description of The National film board of Canada (1985). Here, each film project had to be setup by the appropriate team of cross-disciplinary professionals, forming new adhocracies each time.

Common grounds for programmers and journalists can be described as "boundary objects" (Star and Griesemer 1989). Such objects are interfaces that enable different professional groups to collaborate. It can be as simple as to meet up *via* Teams or showing Power Point-presentations, but also in the form of common language, methods, and ICT tools. Understanding data structures, for example through spreadsheets, the structure of a JSON object, or putting points of data on a map or chart for everyone to study in the ongoing investigation, appear just as important as the coding itself. If journalists and ICT specialists fail to find common ground through boundary objects, they may not succeed in their collaboration, ultimately leading to confrontation or "a clash" between the professions.

## *How is Programming Valued in the Field of Journalism*

Introducing new methods in the context of investigative journalism is a weighted criteria for the SKUP jury, as described in their charter (SKUP 2013). Thus, using programming as a new and efficient tool, one should increase the chances to be awarded with a prize or diploma. Winning candidates for journalism awards are not suitable to explain general trends, but they serve to identify practices that are exemplary and models to emulate in the professional field of journalism (English 2002; Wahl-Jorgensen 2013).

Sub-question three is not given the same attention as the two former. But a simple evaluation of how the field of journalism values programming is useful and valuable. Thus, I look at how projects referring to extensive use of programming are represented amongst the winners of the main SKUP Award and the several diplomas of recognition that are awarded each year.

## Method

The main method used in this study is an open inductive mapping of the 109 methodological reports that contain descriptions of coding. In addition, group interviews are carried out with seven of the teams that used coding as an essential part of their project. The role of the interviews was to clarify factors that were not described well enough in the reports and to expand arguments and explanations, particularly about collaboration and learning.

To isolate the reports containing descriptions of coding, a set of 22 words associated with coding were used in a text search. These words span from general terms such as "program*, code*, algorithms*" along with the names of specific languages, like "python, C++, RStudio, NodeJS" and concepts such as "web scraping, back-end and server-side." Although web scraping can be done manually, initial research show that in this context, web scraping is used as a way to automate harvest from web sites, usually conducted by scripts.

The qualitative analysis software NVivo was used to find the context in which these words were used. Nodes of texts were labelled in accordance with a list of categories linked to the research questions. These categories describe for example arguments for coding, stages in the work process or purpose ("harvest," "building data-set"), descriptions of collaboration, learning or concrete languages and tools. The same pieces of text could be assigned several labels since the categories are not mutually exclusive. NVivo is suited to both counting the occurrence of words, for example how often a certain programming tool is mentioned, and to collecting parts of the text that can provide a better understanding when read and analyse it across several documents. This is done by labelling the excerpts by categories (called "nodes" in NVivo). In accordance with stepwise deductive-induction (SDI) (Tjora 2018), which means that the categories are flexible and can be adjusted according to the findings during the research process. When all the data is sorted into such categories, all texts in all categories are scrutinised in more detail.

The journalistic methods in the data are typically explained semantically by the way in which the sentence is composed, for example: "We used Python to create a script what could scrape the websites and create a CSV-file." This description includes subjects, or actors, performing an action by using a tool, all easily recognisable by the composition of the sentence. It is often necessary to label more than one sentence at once to understand the context. Descriptions of journalistic methods and coding in the SKUP reports span from being very short and superficial to very long and rich.

A fellow researcher has investigated a sample of 10 reports for comparison to secure cross-coder integrity.

The analysis of the reports was important for designing the final interview guide because certain questions were not answered satisfactorily in accordance with all the research questions, particularly those concerning collaboration and learning. To supplement shortcomings in the descriptions in the SKUP reports, seven of the reports with the most extensive descriptions of coding were strategically chosen for group interviews. These teams represented some of the largest news outlets in Norway: the national newspapers VG and Dagens Næringsliv, the documentary series NRK

Brennpunkt, the regional newspapers Bergens Tidende and Adresseavisen, and the niche outlets Klassekampen and Kommunal Rapport.

The interview guide was divided into four sections. First, the group was asked to explain how and why coding was used in the SKUP project the interview was based on, then how programming can be used to solve problems in journalism in general, before going on to explain how the team collaborates when coding, and finally how they learned coding. They were also given an opportunity to provide additional information about other projects or more general thoughts about coding.

It was important to avoid biased questions that could introduce false answers. Claims about poor communication, lack of knowledge etc. was avoided. The questions were mostly open but focused, such as "Why did you use coding in this project?," and "Can you explain how you learned to code?." The answers were transcribed and coded in NVivo and analysed in the same matter as the descriptions from the reports.

The storage of personal data from the interviews was approved by the Norwegian Centre for Research Data (NSD/SIKT).

## Findings

### *Arguments for Programming and the Tools and Languages Used*

The first references to coding appear in projects dating back to 2008, but the number increase every year up to 2019, where a third of all projects refer to coding in some extent (see Table 1). The tools are typically JavaScript (JS) along with PHP and CSS, sometimes mentioned along with JQuery, Angular and Node.js. Adobe Flash is referred to in several project before 2010. Another obsolete tool mentioned is Google Fusion Tables, especially connected with creating interactive maps. This did not require programming skills, but as the Leaflet-library for JS is used as a replacement in newer projects, coding skills have become more important when making digital map presentations. After 2015, Python is increasingly the preferred programming tool, often combined with the Pandas library. RStudio is mentioned in four projects.

**Table 1.** References to coding, absolute and relative to total number of pages per year.

| Year | No.of referances to coding | Relative number of ref/ pages | Dominant tools |
|---|---|---|---|
| 2008 | 2 | 0.29 | Adobe flash |
| 2009 | 5 | 0.69 | |
| 2010 | 3 | 0.29 | SQL, Adobe Flash, API |
| 2011 | 6 | 0.69 | PHP, SQL, Coding (unspecified) |
| 2012 | 12 | 1.40 | Coding (unspecified) |
| 2013 | 17 | 2.25 | JS, PHP, Coding (unspecified) |
| 2014 | 23 | 1.79 | JS, Leaflet, Coding (unspecified) |
| 2015 | 15 | 1.97 | Python, JS, API, Coding (unspecified) |
| 2016 | 18 | 2.35 | Several tools used, none dominate |
| 2017 | 20 | 2.17 | JS, several tools used, none dominate |
| 2018 | 24 | 3.17 | Python, SQL, API, others |
| 2019 | 30 | 3.79 | Python, JS, Coding (unspecified), others |
| 2020 | 24 | 3.25 | Python, JS, others |
| 2021 | 40 | 3.69 | Python, API, Pandas, others |
| Sum | 239 | | |

Table 1 shows references to coding in the SKUP projects. Column 2 shows absolute number of references, where some reports may contain several references. Column 3 shows references compared to the total volume of text in all reports per year.

Table 2. Number of descriptions of coding in different stages of the journalistic process.

| Stage | No. of projects | Amount (words)* | Share (amount) |
|---|---|---|---|
| Harvest | 32 | 4,409 | 27% |
| Building dataset | 11 | 943 | 6% |
| Cleansing | 10 | 933 | 6% |
| Analysis | 50 | 7,315 | 44% |
| Presentation | 30 | 2,977 | 18% |
| Sum | 133 | 16,577 | 100% |

*Metadata with headings from NVivo included.

Table 3. Arguments for choosing coding over other digital tools.

| Argument for coding | Evidence in the projects |
|---|---|
| Automation | Strong |
| Tasks that would otherwise be impossible to do | Strong |
| Fact checking and quality control of data | Mediocre |
| Abstractions: sense-making and new ideas | Weak |
| Visualisation | Weak |
| Crowdsourcing or other ways of including interlopers | None |

Many of the descriptions of coding are unspecified and superficial. There are only thirty projects where programming is described as playing a crucial role.

Coding is described in all parts of the journalistic process, although most of the descriptions are about the harvesting and analysis of data (Table 2).

In line with the expectations of Flew, Spurgeon et al. (2012) Anderson (2013), Diakopoulos (2015), Zamith and Haim (2020), most descriptions are about building algorithms from scratch, and these are utilised in solving tasks in all parts of the work process. There are also some occasions where the journalists use open-source libraries, like Pandas for Python, or find algorithms online, for example on Github, which they then alter. Such reuse and participation in open-source code is as anticipated by Coddington (2015) and Simon (2021). There is a conspicuous lack of descriptions about using programming for visualisation. This may be explained by previous studies of the SKUP reports showing that the journalists give less attention to the presentation and publication of the projects.

The reasons for utilising coding are also much in line with the expectations, at least for the first three categories (Table 3):

According to the descriptions, the main purpose of coding is to help the journalists to handle a lot more data than would be possible manually (Parasie and Dagiral 2013). In some projects, the journalists claim that their work would be impossible without being able to code. The data are either too big, cross-referred or handled in other ways. Also, repeated tasks are often handled by scripts. In the harvesting phase, many projects use web scraping scripts, typically coded with Python, to collect huge amounts of data. The few descriptions of using RStudio mainly concern organising and securing the integrity of multiple datasets. Spreadsheets would be suitable for handling such datasets one-by-one, but RStudio enables several similar datasets to be collected, cleansed and analysed as one, keeping track of all steps of the process as algorithms. This is typically an extension of the digital tools described by Parasie and Dagiral (2013) which may not have been possible without a tool that combines spreadsheets with algorithms. One of the respondants does a simular solution by implementing Visual Basic, which is a programming tool integrated in Microsoft Office.

Parasie and Dagiral (2013) also point out the use of coding for fact-checking and to ensuring the integrity and quality of their data. When asked whether it is hard to detect errors in the scripts, the interviewees say that it is not. Even though it is hard or impossible for other team members and editorial leaders to review the codes themselves, they instead review the results of the scripts and compare them with the raw data manually. Some of the codes are shared on GitHub, and several of the journalists argue that this increases transparency and quality, since they lack a method to peer review their codes in-house. Using code to visualise big data as part of the investigation is considered useful by several of the respondents, particularly to create common ground between the journalists and the coders.

A special case concerning the use of SQL code was found in VG's project "The consequences of the health reform," where the journalists revealed that hospitals transport older adults at night, sometimes dumping them at their house or a nursing home without proper clothing, medicines, or documentation. These transports where logged, but when enquiring about the public records for these logs, the hospitals claimed they would be impossible to obtain and disclose. However, in one of the answers, the journalists found lines of SQL code that had been used to query the hospital's internal database. The journalists recognised the code as SQL and used it to instruct other hospitals to query their database again, using the sampled SQL code. With such a request, they could no longer claim it would be impossible to collect the data, and VG eventually got what they asked for.

In the same project, one of the journalists also points out the rhetorical value of collected and analysed data with high integrity:

> "It is indisputable – [the data] is what is it. They (the hospital managers) cannot claim that they do not transport the older adults home at night." (Respondent M).

In the Adresseavisen project "Møllenberg for rent" (Berdal et al. 2019), the journalists investigated how much of a district in Trondheim was inhabited by students renting flats rather than the owners living there themselves. Their hypothesis was that real estate moguls were transforming the entire district into rentable apartments for students, changing the environment there in an unwanted direction for the rest of the population. The team collected property data along with data on owners and inhabitants and used programming to cross the data for matches. Respondent O, an experienced reporter who has no programming skills explains: "My idea of investigating this was to go from door to door and check who lives there." Her colleague, respondent N, replied that such a solution would be too time consuming, and that choosing this procedure would put an end to the project. So, they found a solution by querying a dataset using code instead.

There are also narratives about episodes when coding had been attempted but abandoned, because it had turned out not to be the best solution. This typically happened when the data was too dirty to create scripts that where timesaving. In the project "Violence at nursery homes," a journalist from Aftenposten wrote: "We had so many different [file] formats of various quality that, in the end, it was more efficient to go through the deviations manually." (Dommerud et al. 2020).

There are no descriptions of crowdsourcing or collaboration with hackers as described by Boyles (2020) and Lewis and Usher (2014). This does not mean that crowdsourcing and such collaboration is not present in the SKUP material (Strømme 2020) only that programming was not used along with programming as a method.

## Collaboration and Learning

The data show that both journalists and ICT specialists use programming as part of the journalistic process. Sometimes, this is explicitly described, such as "Our developer n made a script…," while in other cases the roles are less clear.

Superficial descriptions, such as "this job was done with a Python script" (Kjelleberg et al. 2020), indicate that the author of the report either does not know much more about the tool, or that he thinks it might not be important to further elaborate on it. Either way, this suggests that the detailed coding was left to someone outside the field of journalism, or at least someone who did not participate in documenting the project for the purpose of the SKUP Award. At the same time, such short descriptions give the impression that resolving the given task with coding was unproblematic, and not relevant for elaborate depiction in the methodological report. This may simply be a matter of efficient role distribution, as seen in a machine bureaucracy (M. Weber 1958). The team at Adresseavisen say in their interview that their developer is part of the journalistic environment, being involved full-time in different journalism projects. He fits the description of a "editorial friendly coder" (Stencel and Perry 2016) or "intraloper" (Holton and Belair-Gagnon 2018), even though he spends his workdays as a full member of the editorial staff, as he has the qualifications and background of a professional coder.

Ten such "editorial-friendly coders" are identified in the reports and are referred by titles such as "developer," "programmer" or similar. They sometimes help the code-friendly journalists, but most often, they do the coding themselves without much of their work being included in the SKUP reports.

In about half of the projects with coding, the descriptions of coding are quite rich. These data provide good explanations of the reasons for and aims of the programming and may even include lines of code in details. Such descriptions give an impression of the extent to which the journalists have ownership of the coding themselves, being what Stencel & Perry call "code-friendly journalists." One such example can be seen in VG's "The neglected bridges": "We made SQL queries in our own database, where we had collected our data on the bridge inspections. The queries were built up with "if"-sentences, where certain intervals of inspections were used as conditions (if the condition is fulfilled, it returns something; otherwise, it returns something else). We made chains of several versions of queries (…)" (Engan, Normann, and Nilsen 2018). These descriptions show that the author has practical skills in programming, as a "code-friendly reporter." However, who did the coding and how the coder collaborated with the rest of the team is not described.

When asked to elaborate on this in the interviews, all the teams except one (Adresseavisen, which had an editorial-friendly coder in the team) had journalists who did the coding themselves. All these teams depict difficulties in collaborating with developers/ICT specialists, especially if they were organised as an external department, away from the journalists' workspace. The respondents explain that developers are in short supply and that they often do not understand the needs, perspectives, and time scope of the journalists. Respondent K says: "I couldn't wait a month for a solution. (…) They didn't do what I needed. (…). I believe those computer operating people view us as a plague and a nuisance." Respondent M claims that she asked for ICT help over a year into the project that would later win the SKUP Award: "My bosses bristled against the project for two years. (…) There was a classic clash between two different schools, or ways of doing journalism." She finally received help from a colleague who was about to learn RStudio, but her colleague's frustration was much the same: "When there was a mismatch between the person working with the data and the editorial leader, who does not understand the work, they will not see how demanding and time consuming it was. I failed to communicate how difficult it was" (Respondent N). She spent up to 18 h a day at work, learning to code in RStudio.

In Dagens Næringsliv, the journalists used a programmed tool created by a subsidiary company to performing financial analysis of companies and ownership. The journalist had access to the same tools as the audience (newspaper readers), but they had an API option making it faster to "talk" to the platform. They supplement this by using a journalist with coding capabilities, enabling them to find quick solutions to their on-going journalism work. In their case, the advanced programming was done by another department with which they had very limited collaboration. Although these tools seem to be very significant for the reporters, there was little collaborations between the team and the subsidiary company for further development or help for specific projects.

Such difficulties in terms of collaboration could be caused by failure to find common ground, as referred to as boundary objects by Star and Griesemer (1989) and is also inline with how these professions are described by Stavelin (2014).

## Autodidact Learning

In the smaller outlets, there are few or no developers available. In Klassekampen and Kommunal Rapport, the journalists learned some programming skills and set about the job themselves (Kjelleberg et al. 2020) (Engan, Normann, and Nilsen 2018).

In those projects where the teams need to find their own solutions for coding, they organise their work to fit the given project in an ad hoc manner (Kärreman, Sveningsson, and Alvesson 2002; Mintzberg 1979; Mintzberg and McHugh 1985). The descriptions of journalists performing programming tasks are sometimes introduced with some explanations of how they learned to code. They get introduced to coding though attending a conference, and NICAR is mentioned in several reports. They may also have found a journalistic project that was well suited for programming (for the reasons explained previously). Many journalists subsequently continue their learning and are thus autodidact. They find tutorials on the internet, watch videos on YouTube and Google for answers in Stack Overflow. Another good starting point described is finding and reusing code, for example on GitHub or just through internet searches.

Having a mentor who you can turn to if you get stuck is considered important, but not necessary for everyone. However, the respondents emphasise the importance of being stubborn and willing to sacrifice their spare time to experiment with coding. It is very helpful for the journalists' motivation to have a project that has clearly expressed problems that appear solvable with coding. After solving these problems, the journalists may have a "eureka experience," opening their minds to how programming can be used to solve a broad range of similar tasks: "You realise that everything is just “if”-sentences and “for”-loops” (Respondent L); and: “If you know one language, it will be more efficient to go on with the same language in 98% of the cases” (Respondent F).

The journalists therefore prefer to stick to one programming language, as seen with respondent M, who chose to continue using Microsoft Visual Basic for applications (VBA). Both he and the other respondents said that they would not waste time learning new languages when the one they use seems to solve any task they are up to. Thus, they spend time updating their skills in this specific language rather than reorientating themselves. This may not be the case for ICT specialists and editorial-friendly coders, who may switch between several languages that fit the task at hand (Respondent P).

One of the respondents had a bachelor’s degree in computer science before becoming a journalist, but claims that this was useless in relation to editorial work: “It was a different world. (…) I had to unlearn everything and start learning all over again. (…)” (Respondent K). He and his colleague started to code front-end only to avoid being dependent on their developer’s involvement in the servers.

All the interviewees agree that not all journalists need to learn how to code, but that it is important that someone in the team possesses these skills. Thus, having a development department located away from the team’s workspace is not satisfactory. There is a need for coders as well as journalists being able to recognise code and when programming should be used in the team.

### *Coding is No Recipe for Winning*

At first glance, coding seems to be an important element of winning the main SKUP Award; half of the winners from 2012 through 2021 refer to coding in their reports. However, when examining this in detail, only two of these winning projects state that coding had a crucial role (“The Tidal case,” 2018, Dagens Næringsliv and “The consequences of the health reform,” 2019, VG). However, the jury did not point out the role of programming as one of their weighted reasons for their award decision in neither of these projects (Olufsen 2020; Waatland 2019). In fact, reports that focus a lot on coding are underrepresented in the list of winners and diplomas of recognition. Only two diplomas were awarded among the thirty projects containing most coding, while there are normally four to six diplomas handed out among 40 to 60 candidates every year.

## Discussion

The coding languages and tools described are all in line with the expectations. They are either open-source, or free and easily accessible, and Python becomes the preferred language for most of the teams in recent years. It is remarkable to see the lack of references to programming before 2008, and that it is not until 2012 that the number of references exceeded 10. Thus, coding is not an important part of the workflow in the dataset until recent years, and even then, both the number of references and the number of projects having programming as a crucial part of the project is low (only 30 overall).

The purpose of much of the coding is harvesting, cleansing and simple analysation of the data. Methods of crowdsourcing and collaboration with interlopers are not mentioned in connection to this. A quick reference to the rest of the SKUP material, however, shows that both crowdsourcing and collaboration with interlopers (even hackers) appear in other projects, but programming is not mentioned at all as tools in these contexts.

Much of the coding is conducted in the analysis phase of the workflow, yet the teams do not show any strong evidence of using code for abstraction of data. The main purpose of the analysis for these teams is to filter and query the dataset, and not to use algorithms to discover hidden patterns, nor the use of machine learning. The teams seem to process and understand big data in much the same way as they would treat smaller data sets by using spreadsheets, except that they now use scripts to filter and create subsets instead.

Such use of coding may be seen as trivial. Nonetheless, in many projects, the teams strongly argue that they would not be able to do their investigations if they were unable to code. Thus, these coding solutions they have chosen may prove to be very efficient and important for these projects.

When the coding is to be implemented, the teams either collaborate with ICT specialists, learn coding themselves or a combination. There is a clear distinction between projects where coding is recited as unproblematic and simple, and others, where coding is difficult because the journalists fail to collaborate with the developers. The consequence is that the team of journalists learn coding themselves. These two ways of implementing code fits the descriptions of a bureaucracy and adhocracy, as illustrated below (Figure 1):

In the upper left corner, the collaboration may be seamless, much in the way described in Weber's concept of professional bureaucracy. This is the case in Adresseavisen's "Møllenberg for rent" (Berdal et al. 2019). In the interviews, the

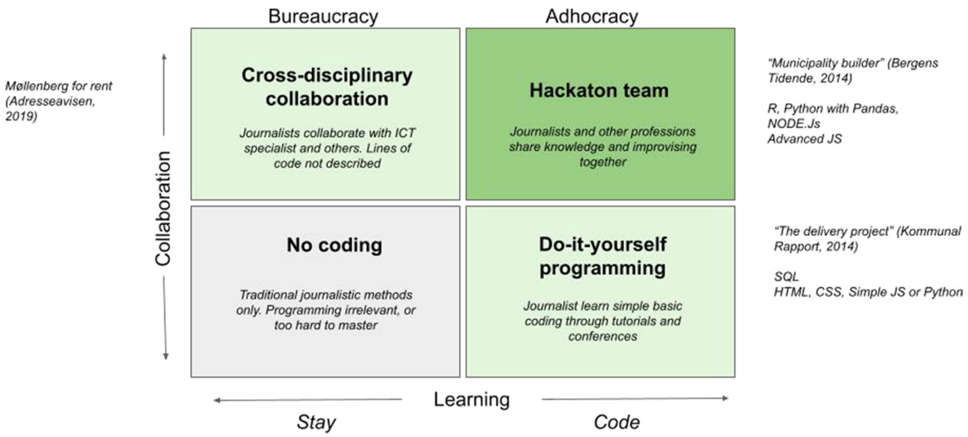
Bureaucracy
Adhocracy
Møllenberg for rent (Adresseavisen, 2019)
Cross-disciplinary collaboration
Journalists collaborate with ICT specialist and others. Lines of code not described
Hackaton team
Journalists and other professions share knowledge and improvising together
"Municipality builder" (Bergens Tidende, 2014)
R, Python with Pandas, NODE.Js Advanced JS
Collaboration
No coding
Traditional journalistic methods only. Programming irrelevant, or too hard to master
Do-it-yourself programming
Journalist learn simple basic coding through tutorials and conferences
"The delivery project" (Kommunal Rapport, 2014)
SQL
HTML, CSS, Simple JS or Python
Learning
Stay
Code

**Figure 1.** Different ways of resolving programming. The lower left corner represents projects where there is no coding. This is the vast majority of the SKUP reports. The upper left corner shows projects where developers/ICT specialists handle all the programming. The boxes to the right represent projects where journalists code themselves, either alone (lower box) or together (upper box). Most of the SKUP reports describing coding to play a crucial role can be placed in the lower right corner, categorised as "Do-it-yourself Programming."

respondents in Adresseavisen agree that with a dedicated coder (described as editorial-friendly), the rest of the team could concentrate on their part of the project, performed their different roles, within normal working hours, and that there was no need to learn new tools.

Just as often, the developers seem to be less friendly, being hard to reach or collaborate with, making the journalists decide to learn coding themselves. This is the case for most of the reports with a high focus on coding. When going about this, the journalists learn some degree of coding that enable them to perform a whole range of problem-solving throughout the entire journalistic process. The amount of improvisation in these projects is immense, and the journalists collaborate well within their teams, typically consisting of between two to four and sometimes five people. The rest of the team may be specialists in other fields, such as economics, have a lot of experience and sources, or have other complementary skills. Although the rest of the team seldom understands coding, they seem to work very well and efficient with their coding-friendly journalist(s).

In some rare occasions, the interest and competence for programming may affect the entire team, enabling them to create more advanced applications of programming together. These people work together as a successful "hackathon team." This description could be applied to the team at Bergens Tidende, where a small team of two coder-friendly journalists worked together. Their solution would be used by other colleagues as well, at a time with much political discussion about merging municipalities. They developed an interface making it possible to interactively merge municipalities on the fly, showing new border and key statistics. These new maps and statistics would then serve as illustrations and fact boxes in a range of stories created by many other journalists in the outlet.

Most of the project, however, can be categorised with "no coding," clearly showing that programming does not need to be an important part of an investigative project.

## Collaboration and Clash of Professions

Although there are good examples of beneficial and efficient cross-disciplinary collaboration between the professions, there is a remarkable distance between what is shown in the advanced digital features of the outlet's websites and how the journalists manage to benefit from this competence during the earlier stages of the work process. There appears to be a great potential for better collaboration between programmers and journalists in these outlets.

Because of the above, journalists often prefer to code independently of server solutions, focusing on front-end solutions and analysing their data locally.

It is also notable that the descriptions of coding carried out by the journalists themselves do not seem to be valued highly by the SKUP jury, as few of these reports are given awards or diplomas. The journalists' enthusiasm to learn and use coding could be interpreted as a movement and an expansion of the field of journalism. Yet they fail to achieve status in the competition, thus not becoming the models to emulate in the professional field of journalism as described by Wahl-Jorgensen and Hanitzsch (2019) and English (2002). This undermines the argument that leaning to code is a necessity for all journalists (Kunert et al. 2022; E. M. Rogers 2003), and this impression is enhanced by those projects that describe successful cross-disciplinary collaboration with programmers.

Yet, there are other arguments for the importance of journalists carrying out the programming work themselves. Most important is that of the eye of the coder-friendly journalist, who can view data in a way that ICT experts cannot. They may see important potential stories in the data because of their knowledge of coding and how data can be inquired. However, identifying possibilities in the data may be a different skill than the actual programming and easier for most journalists to pick up. Most news outlets could benefit the most from following this track, ensuring that investigative teams have easy access to someone who is able to understand data in a journalistic sense, and then either do the coding themselves or assign the tasks to editorial-friendly coders. Journalists considering learning to code should see whether this role lack in the team. if so, they could see a great increase in return of investment by filling it.

From there, whether the journalists chose to code themselves or find themselves appropriate partners in the ICT department, both could be good alternatives. If they choose to learn coding, however, the journalist must be prepared for hard work, improvisation, long hours and perhaps some frustration.

## Conclusion

About one third of the investigative projects nominated for the SKUP Award use programming as a method in the work process in the last five years. Between 2012 and 2021 Programming plays a crucial role in about 15 per cent (30 projects) in total. When programming is used, both journalists and ICT specialists code as part of the process. Programming is described when there is a need to collect and analyse large amounts of data and to automate tasks. In many projects, the journalists argue that their work could not be done without coding. Coding is also used to save time and for quality checking, ensuring that the data is cleansed and analysed in the exact same way throughout, using a script that can be reviewed at any time.

Some projects show a remarkable lack of cross-disciplinary collaboration between journalists and ICT specialists. Some of the journalists claim that the developers do not understand the journalists' scope and perspectives, making them unable to see important factors or possibilities. They may be unable to provide solutions on time or even at all. Thus, there appear to be great potential for better collaboration between programmers and journalists in many cases.

If the collaboration between journalists and ICT experts fails, or there are no ICT experts to ask, some of the journalists learn coding themselves. This typically starts by attending a short course or presentation at a conference, such as NICAR, sacrificing a lot of spare time to learn to code by themselves, following tutorials online and searching for answers. It is helpful to have a journalistic project that is suitable for coding and to have a mentor. The journalists stick to one programming language, typically Python, but sometimes JavaScript or SQL, and in rare occasions RStudio. The journalists learn a few but efficient tools for programming in a "do-it-yourself" manner.

The highest return of investment for learning to code is when the team acquire a person in the role of a "code-friendly journalist," building a bridge between the professions. Alternatively, the journalists can learn data structures and how data can be inquired to accurately ask for help by "editorial-friendly coders."

The journalists themselves nevertheless argue that it is important that some, but not all, journalists know how to program since this will enable them to read data and see possibilities in different ways than a developer would. In any case, using programming as part of a project nominated for the SKUP Award does not appear to enhance the chances of winning.

## Disclosure Statement

No potential conflict of interest was reported by the author(s).

## References

Abbott, A. 2014. *The System of Professions: An Essay on the Division of Expert Labor*. Chicago: University of Chicago press.

Adegboyega O., and B. Heravi. 2017. "Patterns In Award Winning Data Storytelling - Story Types, Enabling Tools and Competences." *Digital Journalism* 6(5):1–26. https://doi.org/10.1080/21670811.2017.1403291

Ananny, M., and K. Crawford. 2015. "A Liminal Press: Situating News App Designers within a Field of Networked News Production." *Digital Journalism* 3 (2): 192–208. https://doi.org/10.1080/21670811.2014.922322

Anderson, C. W. 2013. "Towards a Sociology of Computational and Algorithmic Journalism." *New Media & Society* 15 (7): 1005–1021. https://doi.org/10.1177/1461444812465137

Appelgren, Ester, and Gunnar Nygren. 2014. "Data Journalism In Sweden." *Digital Journalism* 2 (3): 394–405. https://doi.org/10.1080/21670811.2014.884344

Berdal, M., G. Holstad, I. Meisingset, J. Nilsson, E. Rasmussen, and A. Ødegaard. 2019. Møllenberg til leie - Metoderapport til SKUP 2018. Adresseavisen. https://www.skup.no/sites/default/files/metoderapport/2019-02/skup-mollenberg.pdf

Bjerknes, F. 2020. "Inventive Factfinders: Investigative Journalism as Professional Self-Representation, Marker of Identity and Boundary Work." *Journalism Practice* 16 (6): 1037–1056. https://doi.org/10.1080/17512786.2020.1845780

Bounegru, L., L. Chambers, and J. Gray. 2012. *The Data Journalism Handbook*. Sebastopol: O'reilly.

Bowers, D., S. Rogers, and J. Schwabish. 2017. Data Journalism 2017 - The current state and challanges facing the field today. Google News Labs. https://issuu.com/saladeprensa/docs/data_journalism_in_2017_45e68ba08293b9

Boyles, J. L. 2020. "Laboratories for News? Experimenting with Journalism Hackathons." *Journalism* 21 (9): 1338–1354. https://doi.org/10.1177/1464884917737213

Britannica, T. Editors of Encyclopaedia (2022, December 6). Computer program. Encyclopedia Britannica. https://www.britannica.com/technology/computer-program

Carlson, M., and S. C. r Lewis. 2015. *Boundaries of Journalism - Professionalism, Practices and Participation*. In. New York: Routledge.

Carson, A., and K. Farhall. 2018. "Understanding Collaborative Investigative Journalism in a "Post-Truth" Age." *Journalism Studies* 19 (13): 1899–1911. https://doi.org/10.1080/1461670X.2018.1494515

Coddington, M. 2015. "Clarifying Journalism's Quantitative Turn: A Typology for Evaluating Data Journalism, Computational Journalism, and Computer-Assisted Reporting." *Digital Journalism* 3 (3): 331–348. https://doi.org/10.1080/21670811.2014.976400

Davies, M. 2005. "Do It Yourself: Punk Rock and the Disalienation of International Relations." In *Resounding International Relations*, 113–140. New York: Springer.

Deuze, M. 2005. "What is Journalism? Professional Identity and Ideology of Journalists Reconsidered." *Journalism* 6 (4): 442–464. https://doi.org/10.1177/1464884905056815

Diakopoulos, N. 2015. "Algorithmic Accountability: Journalistic Investigation of Computational Power Structures." *Digital Journalism* 3 (3): 398–415. https://doi.org/10.1080/21670811.2014.976411

Domingo, D., and V. Wiard. 2016. "News Networks." *The SAGE Handbook of Digital Journalism*, 397–409. Croydon: Sage.

Dommerud, T., J. G. Furuly, K. Nipen, T. J. Strøm, and N. S. Torset. 2020. Sykehjemsvolden - Metorerapport til SKUP 2019. *Aftenposten*. https://www.skup.no/sites/default/files/2020-02/Sykehjemsvolden%20-%20metoderapoort.pdf

Engan, Ø., M. G. Normann, and S. Nilsen. 2018. De forsømte broene - Metoderapport til SKUP 2017. *VG*. https://www.skup.no/sites/default/files/metoderapport/2018-03/Metoderapport%20-%20De%20fors%C3%B8mte%20broene.pdf

English, J. F. 2002. "Winning the Culture Game: Prizes, Awards, and the Rules of Art." *New Literary History* 33 (1): 109–135. https://doi.org/10.1353/nlh.2002.0005

Flew, T., C. Spurgeon, A. Daniel, and A. Swift. 2012. "The Promise of Computational Journalism." *Journalism Practice* 6 (2): 157–171. https://doi.org/10.1080/17512786.2011.616655

Foust, J. C., and K. A. Bradshaw. 2020. "Pushing Boundaries: How Coding Is (and Isn't) Taught in Accredited Journalism Programs." *Journalism & Mass Communication Educator* 75 (3): 335–347. https://doi.org/10.1177/1077695820912953

Furuly, J. G. 1994. "Nordmenn i Tysk Nazi-datanettverk - Metoderapport Til SKUP 1993." *Aftenposten*. Oslo: SKUP.

Gieryn, T. F. 1983. "Boundary-Work and the Demarcation of Science from Non-Science: Strains and Interests in Professional Ideologies of Scientists." *American Sociological Review* 48 (6): 781–795. https://doi.org/10.2307/2095325

Gynnild, A. 2007. "Creative Cycling of News Professionals." *The Grounded Theory Review: An International Journal* 6(2): 67–92.

Haim, M. 2022. "The German Data Journalist in 2021." *Journalism Practice*. Advance online publication. https://doi.org/10.1080/17512786.2022.2098523

Heravi, B. R. 2018. *Data Journalism in 2017: A Summary of Results from the Global Data Journalism Survey*. In *Transforming Digital Worlds: 13th International Conference, iConference 2018,* Sheffield, UK, March 25–28, 2018, Proceedings 13, 107–113. Springer International Publishing.

Holton, A. E., and V. Belair-Gagnon. 2018. "Strangers to the Game? Interlopers, Intralopers, and Shifting News Production." *Media and Communication* 6 (4): 70–78. https://doi.org/10.17645/mac.v6i4.1490

Khazan, O. 2013. Should journalism schools require reporters to 'learn code'? No. The Atlantic

Kjelleberg, E., A. Moland, L. K. Noer, and R. Solvang. 2020. Trigger Warning - Metoderapport til SKUP 2019. *NRK*. https://www.skup.no/sites/default/files/2020-02/Trigger%20Warning.pdf

Klein, G., B. Moon, and R. R. Hoffman. 2006. "Making Sense of Sensemaking 1: Alternative Perspectives." *IEEE Intelligent Systems* 21 (4): 70–73. https://doi.org/10.1109/MIS.2006.75

Krumsvik, A. H., S. Milan, N. N. Bhroin, and T. Storsul. 2019. "Making Media: Production, Practices, and Professions." In D. Mark & P. Mirjam (Eds.), 14. *Making (Sense of) Media Innovations* (s. 193–206): Amsterdam University Press. https://doi.org/10.1017/9789048540150.014

Kunert, J., J. Frech, M. Brüggemann, V. Lilienthal, and W. Loosen. 2022. "How Investigative Journalists Around the World Adopt Innovative Digital Practices." *Journalism Studies* 23 (7): 761–780. https://doi.org/10.1080/1461670X.2022.2033636

Kärreman, D., S. Sveningsson, and M. Alvesson. 2002. "The Return of the Machine Bureaucracy?-Management Control in the Work Settings of Professionals." *International Studies of Management & Organization* 32 (2): 70–92.

Lewis, S. C. 2012. "The Tension between Professional Control and Open Participation: Journalism and Its Boundaries." *Information, Communication & Society* 15 (6): 836–866. https://doi.org/10.1080/1369118X.2012.674150

Lewis, S. C., and N. Usher. 2013. "Open Source and Journalism: Toward New Frameworks for Imagining News Innovation." *Media, Culture & Society* 35 (5): 602–619. https://doi.org/10.1177/0163443713485494

Lewis, S. C., and N. Usher. 2014. "Code, Collaboration, and the Future of Journalism: A Case Study of the Hacks/Hackers Global Network." *Digital Journalism* 2 (3): 383–393. https://doi.org/10.1080/21670811.2014.895504

Loosen, W. 2021. "49. Data-Driven Gold Standards: What the Field Values as Award-Worthy Data Journalism." In *The Data Journalism Handbook: Towards a Critical Data Practice*, edited by L. Bounegru and J. Grey, 360–369. Amsterdam: Amsterdam University Press.

Marino, M. C. 2020. *Critical Code Studies*. Cambridge, MA: MIT Press.

Mintzberg, H. 1979. *The Structuring of Organizations*. Englewood Cliffs, NJ: Prentice Hall.

Mintzberg, H., and A. McHugh. 1985. "Strategy Formation in an Adhocracy." *Administrative Science Quarterly* 30 (2): 160–197. https://doi.org/10.2307/2393104

Olufsen, B. 2019. [Kriteriene for SKUP-prisen].

Olufsen, B. 2020. Derfor vant VG SKUP-prisen. Oslo. *SKUP*. https://www.skup.no/2020/04/derfor-vant-vg-skup-prisen

Ottosen, R. 1996. *Fra Fjærpenn Til Internett: Journalister i Organisasjon Og Samfunn*. In: Aschehoug. https://www.nb.no/items/c6f1c135f463a905c44ae05b508095a6?page=0&searchText=fra%20fj%C3%A6rpenn%20til%20internett

Parasie, S., and E. Dagiral. 2013. "Data-Driven Journalism and the Public Good:"Computer-Assisted-Reporters" and "Programmer-Journalists" in Chicago." *New Media & Society* 15 (6): 853–871. https://doi.org/10.1177/1461444812463345

Raja, T., and M. Jun. 2014. *Is Coding the New Literacy*. Mother Jones. https://www.motherjones.com/media/2014/06/computer-science-programming-code-diversity-sexism-education/

Rogers, E. M. 2003. *Diffusion of Innovations*, 5th edition. New York: Free Press.

Raaum, O. 1999. *Pressen Er Løs!: Fronter i Journalistenes Faglige Frigjøring*. Oslo: Pax.

Schiffrin, D., D. Tannen, and H. E. Hamilton. 2015. *The Handbook of Discourse Analysis*. West Sussex: Wiley Blackwell.

Shoemaker, P. J., and S. D. Reese. 1996. *Mediating the Message*. White Plains, NY: Longman.

Simon, B. 2021. 18. Coding With Data in the Newsroom. Towards a Critical Data Practice, 124.

SKUP. 2010. *SKUPs Jubileumstidsskrift 2010*. Oslo: SKUP.

SKUP. 2013. SKUPs vedtekter. https://www.skup.no/skups-vedtekter

Splendore, S., P. Di Salvo, T. Eberwein, H. Groenhart, M. Kus, and C. Porlezza. 2016. "Educational Strategies in Data Journalism: A Comparative Study of Six European Countries." *Journalism* 17 (1): 138–152. https://doi.org/10.1177/1464884915612683

Star, S. L., and J. R. Griesemer. 1989. "Institutional Ecology, Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39." *Social Studies of Science* 19 (3): 387–420. https://doi.org/10.1177/030631289019003001

Stavelin, E. 2014. *Computational Journalism - When Journalism Meets Programming*. Bergen: Universitetet i bergen.

Stencel, M., and K. Perry. 2016. Superpowers: The digital skills media leaders say newsrooms need going forward. *Webpage*. [*Accessed 28 July*, 2016] *Available from* http://towknight. or g/research/superpowers.

Strømme, E. S. 2020. "Hva Forteller SKUP-Rapportene Om Norsk Gravejournalistikk?" *Norsk Medietidsskrift* 27 (2): 1–16. https://doi.org/10.18261/ISSN.0805-9535-2020-02-04

Strømme, E. S. 2022. *Kokebok Fra SKUP-Kjøkkenet* (Cookbook Fra the SKUP-Kitchen). Bergen: *Fagbokforlaget*.

Tjora, A. 2018. *Qualitative Research as Stepwise-Deductive Induction*. Oxon: Routledge.

Tuchman, G. 1978. Making news: A study in the construction of reality.

Wahl-Jorgensen, K. 2013. "The Strategic Ritual of Emotionality: A Case Study of Pulitzer Prize-Winning Articles." *Journalism* 14 (1): 129–145. https://doi.org/10.1177/1464884912448918

Wahl-Jorgensen, K., and T. Hanitzsch. 2019. *The Handbook of Journalism Studies,* 3–20. New York: Routledge.

Weber, M. 1958. *Essays in Society*. New York: Oxford University Press.

Weber, M. S., and A. Kosterich. 2018. "Coding the News: The Role of Computer Code in Filtering and Distributing News." *Digital Journalism* 6 (3): 310–329. https://doi.org/10.1080/21670811.2017.1366865

Weber, W., M. Engebretsen, and H. Kennedy. 2018. “Data Stories: Rethinking Journalistic Storytelling in the Context of Data Journalism.” *Studies in Communication Sciences* 18 (1): 191–206. https://doi.org/10.24434/j.scoms.2018.01.013

Wiik, J. 2015. “Internal Boundaries: The Stratification of the Journalistic Collective.” In *Boundaries of Journalism*, 118–133. London: Routledge.

Waatland, E. 2019. SKUP-prisen til Dagens Næringsliv for «Tidal»-saken. Oslo. m24.no. https://m24.no/dagens-naeringsliv-kjetil-saether-markus-tobiassen/skup-prisen-til-dagens-naeringsliv-for-tidal-saken/214587

Young, M. L., A. Hermida, and J. Fulda. 2018. “What Makes for Great Data Journalism?” *Journalism Practice* 12 (1): 115–135. https://doi.org/10.1080/17512786.2016.1270171

Zamith, R., and M. Haim. 2020. “Algorithmic Actants in Practice, Theory, and Method.” *Media and Communication* 8 (3): 1–4. https://doi.org/10.17645/mac.v8i3.3395