No. 2021-1542

United States Court of Appeals for the Federal Circuit

SAS INSTITUTE INC.

Plaintiff-Appellant,

v.

WORLD PROGRAMMING LIMITED,

Defendant-Appellee.

On Appeal from the United States District Court
for the Eastern District of Texas
No. 2:18-cv-00295-JRG, Hon. J. Rodney Gilstrap

BRIEF FOR AMICI CURIAE COMPUTER SCIENTISTS IN SUPPORT OF REVERSAL

Elizabeth Rogers Brannen
STRIS & MAHER LLP
777 S. Figueroa Street, Suite 3850
Los Angeles, CA 90017
Telephone: (213) 995-6800
ebrannen@stris.com

Bridget Asay
Principal Counsel
STRIS & MAHER LLP
28 Elm Street, Second Floor
Montpelier, VT 05602
Telephone: (802) 858-4285
basay@stris.com

Counsel for Amici Curiae

May 21, 2021

UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

CERTIFICATE OF INTEREST

Case Number 2021-1542

Short Case Caption SAS Institute Inc. v. World Programming Limited

Filing Party/Entity Amici Curiae Computer Scientists

Instructions: Complete each section of the form. In answering items 2 and 3, be specific as to which represented entities the answers apply; lack of specificity may result in non-compliance. **Please enter only one item per box; attach additional pages as needed and check the relevant box**. Counsel must immediately file an amended Certificate of Interest if information changes. Fed. Cir. R. 47.4(b).

I certify the following information and any attached sheets are accurate and complete to the best of my knowledge.

Date: 05/21/2021

Signature: 

Name: Bridget Asay

1. Represented Entities. Fed. Cir. R. 47.4(a)(1).	**2. Real Party in Interest.** Fed. Cir. R. 47.4(a)(2).	**3. Parent Corporations and Stockholders.** Fed. Cir. R. 47.4(a)(3).
Provide the full names of all entities represented by undersigned counsel in this case.	Provide the full names of all real parties in interest for the entities. Do not list the real parties if they are the same as the entities. ■ None/Not Applicable	Provide the full names of all parent corporations for the entities and all publicly held companies that own 10% or more stock in the entities. ■ None/Not Applicable
Dr. Laurie Williams		
Dr. Lucas Layman		
Dr. Mark Sherriff		

☐ Additional pages attached

4. Legal Representatives. List all law firms, partners, and associates that (a) appeared for the entities in the originating court or agency or (b) are expected to appear in this court for the entities. Do not include those who have already entered an appearance in this court. Fed. Cir. R. 47.4(a)(4).

☐ None/Not Applicable ☐ Additional pages attached

Bridget Asay Partner, Stris & Maher LLP		
Elizabeth Brannen Partner, Stris & Maher LLP		

5. Related Cases. Provide the case titles and numbers of any case known to be pending in this court or any other court or agency that will directly affect or be directly affected by this court's decision in the pending appeal. Do not include the originating case number(s) for this case. Fed. Cir. R. 47.4(a)(5). See also Fed. Cir. R. 47.5(b).

■ None/Not Applicable ☐ Additional pages attached

6. Organizational Victims and Bankruptcy Cases. Provide any information required under Fed. R. App. P. 26.1(b) (organizational victims in criminal cases) and 26.1(c) (bankruptcy case debtors and trustees). Fed. Cir. R. 47.4(a)(6).

■ None/Not Applicable ☐ Additional pages attached

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes

Other Authorities

INTEREST OF AMICI CURIAE[1]

Amici curiae are computer scientists who share a keen interest in assuring robust legal protection for creative, proprietary computer programs. They believe that intellectual property law should promote innovation and create incentives for investments in new and improved applications and programs. They also know, through their own work and their teaching experience, that designing and developing computer programs is a challenging and fundamentally creative process. Allowing competitors to copy (without permission or payment) expressive elements of proprietary computer programs greatly diminishes the value of that creative work. *Amici* believe that copyright law should recognize and protect original, creative computer programs, as Congress intended.

Dr. Laurie Williams is a Distinguished University Professor in the Computer Science Department of the College of Engineering at North Carolina State University (NCSU). Laurie is a co-director of the NCSU Science of Security Lablet sponsored by the National Security Agency, the NCSU Secure Computing Institute,

[1] Stris & Maher LLP, counsel for amici, authored this brief. Counsel for appellant SAS Institute Inc. and counsel for appellee World Programming Limited consented to the filing of this brief. No counsel for a party authored this brief in whole or in part. No party or counsel for a party contributed money that was intended to fund preparing or submitting the brief. No one other than *amici* and their counsel contributed money that was intended to fund preparing or submitting the brief. *See* Fed. Cir. R. 29(a)(4)(E).

and is the Principal Cybersecurity Technologist of the SecureAmerica Institute. Dr. Williams's research focuses on software security; agile software development practices and processes, particularly continuous deployment; and software reliability, software testing and analysis. She is an ACM and an IEEE Fellow.

Dr. Lucas Layman is an Assistant Professor in the Department of Computer Science at the University of North Carolina Wilmington. From 2009-2017, he was a research scientist at Fraunhofer USA Inc.'s Center for Experimental Software Engineering. Dr. Layman has managed over $1 million in applied software engineering research programs with software organizations including NASA, US Department of Defense, Microsoft, IBM, and Cisco. His research focuses on software engineering, software data analytics, and human factors in computer security.

Dr. Mark Sherriff is a Professor in the Department of Computer Science in the School of Engineering and Applied Science at the University of Virginia. He also serves as the director of the Center for Innovation in Computing Education and Outreach. Dr. Sherriff has won numerous awards for his teaching, including the 2016 IEEE Computer Society Computer Science and Engineering Undergraduate Teaching Award (national professional society award) and the 2014 University of Virginia All-University Teaching Award. His research areas include CS education

and the scholarship of teaching and learning. Dr. Sherriff is a Senior Member of the ACM and IEEE.

SUMMARY OF ARGUMENT

The district court wrongly failed to acknowledge the range of copyrightable elements of the SAS System. The SAS System is a proprietary computer program that businesses, academic researchers, and governments use for data management and analytics. SAS Institute has spent decades and untold resources developing its suite of programs that it licenses to customers all over the world. It is undisputed, and the trial court found, that the defendant World Programming Limited, or WPL, is a competitor of SAS whose "business was to 'clone' the SAS Software." Appx9; *see also* Appx15 (SAS "presented repeated evidence of factual copying"). The court, however, dismissed SAS's infringement claims, concluding that SAS had not shown that its work was protected by copyright.

That result is indefensible. There can be no serious dispute that the SAS System is an original and creative work, and its input formats and output designs are protectable elements of that work. As computer scientists who are familiar with the SAS System, with competing software programs, and with the design of proprietary software generally, we are greatly troubled by the district court's incorrect understanding of computer science and misapplication of copyright law in this case. Allowing wholesale copying of a competitor's proprietary software undermines

innovation, reduces the value of original programs, and disserves the creative development process.

We accordingly urge the Court to reverse, addressing two primary issues.

First, we draw on our expertise in the field to explain why the SAS System is a creative, original work. The SAS System, like other sophisticated, proprietary programs, represents the collective efforts of scores of software designers and programmers. The program's input formats and output designs are the product of their creative decisions.

Input formats are the way a user provides instructions to the software, to guide the mathematical calculations and statistical analyses that the user needs—whether a simple calculation of averages or multivariable regression analyses. SAS created hundreds of procedures called PROCs that are designed to generate different analyses. It named the PROCs, created their organization, structure, and syntax, and decided on options and default parameters for each PROC. The input formats incorporate mathematical formulas but the language and format of the textual inputs are the creation of SAS. As the record in this case shows, other programs have very different input formats for similar statistical analyses. Copyright law should protect the work created by SAS.

The same is true of the program's output designs. Output designs are the results generated by the program, which take the form of text, charts, and graphs of

all kinds. A given PROC may be capable of generating one or two or numerous outputs. Again, the format and appearance of the output designs were created by SAS. SAS made choices about, for example, the arrangement and labeling of data; fonts; colors; size and proportion of layouts; default settings; and the style of graphs and charts.

WPL is free to create a competing program that offers similar statistical tools. Other companies have, including IBM and Microsoft. But WPL should not be free to copy the design that SAS created for its input formats and output designs. It may have an easier time attracting new customers if its product has the same "look and feel" as the SAS System. But that is true of any knockoff product, whether it is high-end sneakers or proprietary software. Copyright law does not permit competitors to make use of others' intellectual property just because it is cheaper and easier to do so.

Second, we explain why copyright protection for original, creative software programs is critical to promoting innovation and creating incentives for the kinds of investments and risks that drive our economy. Proprietary software programs meet important needs for individuals, businesses, scientists, and governments. They represent an important engine for economic growth. And critically, by protecting original works, copyright law pushes competitors to not just copy, but to improve. Instead of making it the same, make it better—and copyright law will protect that

new design too. Legal protection for intellectual property drives the kind of innovation that benefits consumers and the market.

ARGUMENT

I. The SAS System is a creative, original work.

A. The SAS System.

The SAS System is a proprietary, integrated software system that SAS licenses to businesses across the globe. SAS industry clients rely on this suite of business software to, among other things, collect and analyze data, assist with project management and quality improvement, and generate models and forecasts that shape critical business decisions. SAS's customers include 91 of the top 100 companies on the 2020 Fortune 500 list.[2]

SAS is widely used in academic research as well. The SAS System allows users to perform a variety of tasks related to data access, data management, data and statistical analysis, and data presentation. Researchers use SAS products for important pursuits like fighting cancer.[3] A SAS team has partnered with a New York

[2] SAS 2020-2021 Annual Report at 3 (2021), https://www.sas.com/content/dam/SAS/documents/corporate-collateral/annual-report/company-overview-annual-report.pdf.

[3] *See, e.g.*, Alison Bolen, *Analytics Leads To Cancer Cures*, SAS Institute Inc., http://bit.ly/31NO3mq (last visited May 20, 2021).

City nonprofit group on a sophisticated data analysis that reveals racial disparities in home ownership, mortgages, and foreclosure.[4]

As the COVID-19 pandemic has impacted public health, governments, the economy, transportation, manufacturing, and supply chains, data analytics has perhaps never been more critical. Governments and public health officials work with the SAS System to map the outbreak, generate COVID dashboards, facilitate contact tracing, and inform public health guidelines.[5] Healthcare providers like the Cleveland Clinic used the SAS System to model "simulations that estimate[d] worst case, best case and most likely scenarios," including the critical ability to adjust simulations "in real time as the situation and data change[d]."[6] Businesses across the globe have relied on the SAS System to help predict changes in consumer demand and address unforeseen interruptions in supply chains.[7]

As computer scientists, we understand both the value of sophisticated programs like the SAS System and the high degree of creativity involved in

---

[4] *See* Alison Bolen, *It's Personal: SAS data scientists explore racial disparities in NYC housing data*, SAS Blogs (Nov. 20, 2020), https://blogs.sas.com/content/sascom/20201/20/center-nyc/

[5] *See* Jim Goodnight, *Finding COVID-19 answers with data and analytics* (2020), https://www.sas.com/en_us/insights/articles/analytics/finding-covid-19-answers-with-data-and-analytics.html; *see also* SAS 2020-2021 Annual Report at 7 (2021).

[6] *Id.*.

[7] *Id.*; *see also id.* at 21 (2021).

designing them. This dispute centers on SAS's critical effort to protect its original, creative work from wholesale copying by a competitor. SAS has shown—indeed, there is no dispute on this point—that WPL copied input formats and output designs from the SAS System to develop a competing product that is, effectively, a clone. Appx9.

In this brief, we discuss two elements of the SAS System that WPL copied: input formats, and especially SAS's PROCs, and output designs.

Input formats. Users of the SAS System most commonly interact with the SAS System through input formats. Input formats can be thought of as instructions to the software. As SAS's former chief technical officer explained, the SAS System allows a user "to answer a question or series of questions" by processing and analyzing data and then generating "the designed output with concise written instructions that would otherwise require literally hundreds of thousands of lines of code in a low-level programming language." *SAS Institute Inc. v. World Programming Ltd. et al.*, 496 F. Supp. 3d 1019 (E.D. Tex. 2020), ECF Doc. 444, at 20 (hereafter E.D. Tex. Opening Br.).

An example illustrates how this works. Suppose a statistics student is looking at the relationship between baseball players' salaries and performance measures like batting statistics. If this were 1960, the student would need pencil, paper, a slide rule, and a lot of time. In 1975, an enterprising student could try writing computer code.

On *Star Trek*, the student could say: "Computer, please predict the 2022 salary for Mookie Betts based on his batting average and RBI for the last five years." In 2021, using the SAS System, the student is much closer to *Star Trek* than to the slide rule: it's a fast, efficient tool for performing complex calculations, and SAS has already written the underlying code. But the software doesn't speak conversational English (yet). The software is designed to require a textual input format to (1) tell the software what kind of analysis should be done and (2) provide certain information and direction to guide how the analysis is performed.

To enable the student to complete her assignment, the software is designed to use input formats associated with the PROCs. PROCs are central to the SAS System. A PROC is a set of instructions for performing certain calculations and statistical analyses. It corresponds to prewritten, specialized software. Instead of writing the thousands of lines of code that each analysis requires, a user chooses one or more PROCs (from a library of over 500) to generate the desired analysis. The PROCs are organized into categories, such SAS/STAT, and described in SAS manuals and user guides.

This student might start working with a data set using "PROC UNIVARIATE." Using PROC UNIVARIATE, she can perform complex univariate analyses—that is, analyze the data set with respect to one variable, like salary or

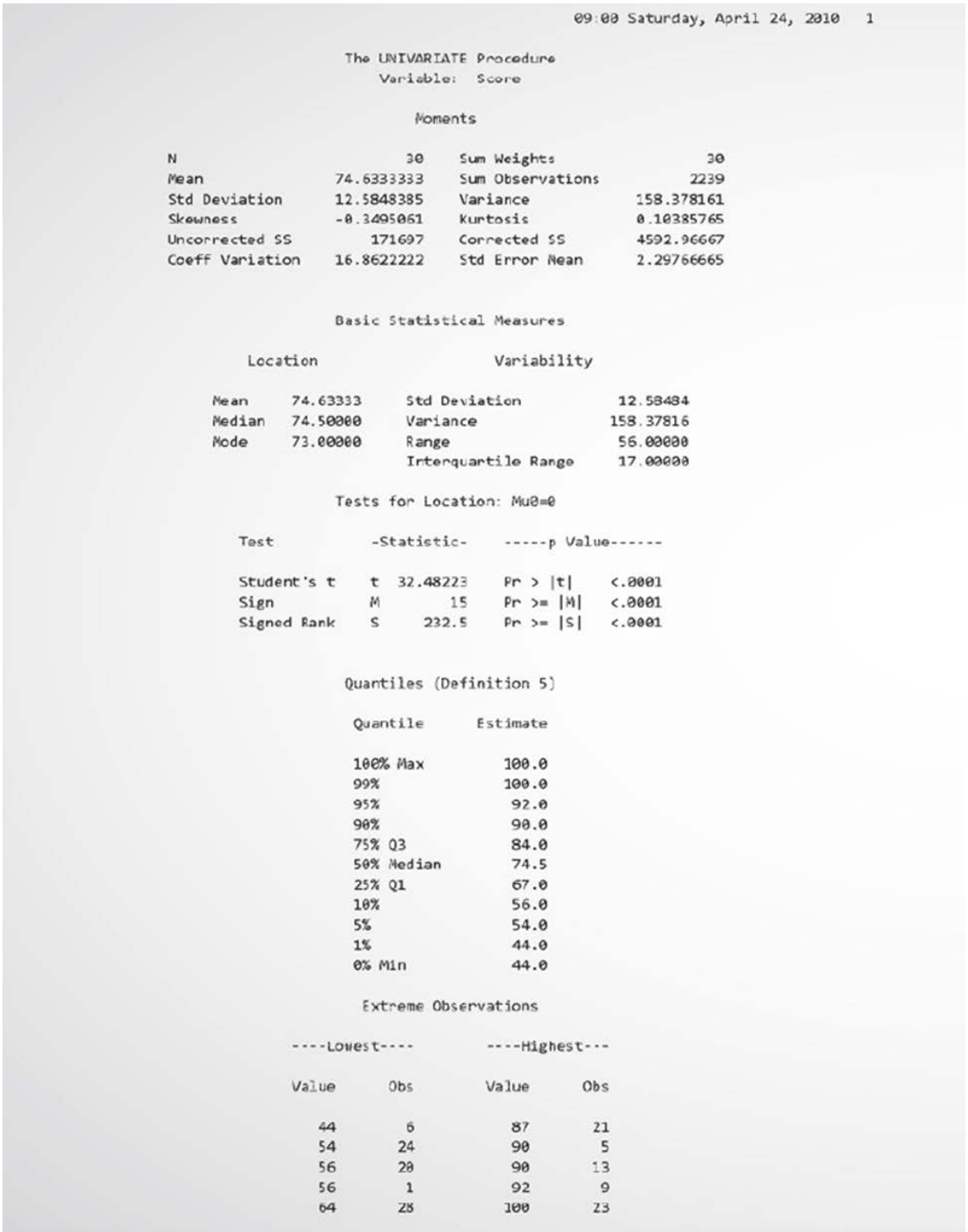
batting average.[8] To do so, she interacts with the System in a particular way, using the input formats for PROC UNIVARIATE to provide information and give direction to the software. Next, the student may want to try modeling relationships between salaries and performance measures. To perform regression analyses, she would turn to PROC REG (for regression). Like PROC UNIVARIATE, PROC REG provides a format for the student to interact with the software, identifying the data set, the variables to include or exclude, and other information necessary to generate the desired results.[9]

Output designs. Output designs, as the name suggests, are the output formats generated by a PROC. They are how the program presents the information and analysis for the user—or, more simply, what the user sees as the results. Each PROC includes output designs, in the form of tables, graphs, or charts, with a particular arrangement and appearance. The outputs are designed by SAS to display the information in a particular order, appearance, and format. The record below includes

[8] PROC UNIVARIATE is described further here: *PROC UNIVARIATE Statement*, Base SAS Procedures Guide (August 12, 2020), https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.5/procstat/procstat_univariate_syntax01.htm.

[9] PROC REG is explained further here, including an example based on baseball players' salaries: *Example 102.1 Modeling Salaries of Major League Baseball Players*, SAS/STAT User's Guide (December 13, 2019), https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_reg_examples01.htm.

a sample output from PROC UNIVARIATE showing an analysis of a data set consisting of 30 student test scores. It looks like this:



09:00 Saturday, April 24, 2010 1

The UNIVARIATE Procedure
Variable: Score

Moments

N	30	Sum Weights	30
Mean	74.6333333	Sum Observations	2239
Std Deviation	12.5848385	Variance	158.378161
Skewness	-0.3495061	Kurtosis	0.10385765
Uncorrected SS	171697	Corrected SS	4592.96667
Coeff Variation	16.8622222	Std Error Mean	2.29766665

Basic Statistical Measures

Location		Variability	
Mean	74.63333	Std Deviation	12.58484
Median	74.50000	Variance	158.37816
Mode	73.00000	Range	56.00000
		Interquartile Range	17.00000

Tests for Location: Mu0=0

Test	-Statistic-		-----p Value------	
Student's t	t	32.48223	Pr > |t|	<.0001
Sign	M	15	Pr >= |M|	<.0001
Signed Rank	S	232.5	Pr >= |S|	<.0001

Quantiles (Definition 5)

Quantile	Estimate
100% Max	100.0
99%	100.0
95%	92.0
90%	90.0
75% Q3	84.0
50% Median	74.5
25% Q1	67.0
10%	56.0
5%	54.0
1%	44.0
0% Min	44.0

Extreme Observations

----Lowest----		----Highest---	
Value	Obs	Value	Obs
44	6	87	21
54	24	90	5
56	28	90	13
56	1	92	9
64	25	100	23

See SAS Opening Brief, Doc. 13, at 22-23 (citing Appx687). If a student used PROC UNIVARIATE to analyze a baseball team's batting averages, the data would be different, but the output format would be the same. The format is part of the design of the System.

A complex PROC like PROC REG can generate a variety of outputs, including tables and graphs. The current SAS online user guide illustrates sample outputs from PROC REG for several different problems.[10] Here is part of an analysis of baseball player salaries and batting statistics from the 1980s:

The REG Procedure
Model: MODEL1
Dependent Variable: logSalary Log Salary

Number of Observations Read	322
Number of Observations Used	263
Number of Observations with Missing Values	59

Analysis of Variance

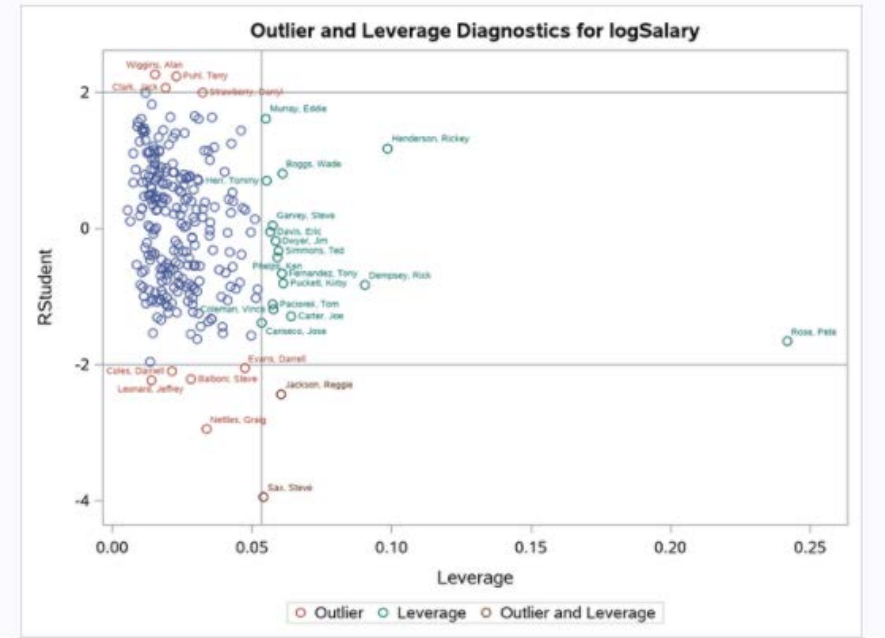
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	121.53052	20.25509	60.56	<.0001
Error	256	85.62322	0.33447		
Corrected Total	262	207.15373			

[10] *See Example 102.1 Modeling Salaries of Major League Baseball Players*, SAS/STAT User's Guide (December 13, 2019).

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	4.14614	0.13612	30.46	<.0001
nHits	Hits in 1986	1	0.00663	0.00210	3.15	0.0018
nRuns	Runs in 1986	1	0.00019890	0.00398	0.05	0.9602
nRBI	RBIs in 1986	1	0.00125	0.00235	0.53	0.5947
nBB	Walks in 1986	1	0.00672	0.00239	2.81	0.0054
YrMajor	Years in the Major Leagues	1	0.07108	0.01925	3.69	0.0003
CrHits	Career Hits	1	0.00023910	0.00014571	1.64	0.1020

As the student works through the data to examine and refine a statistical model, other output designs from PROC REG will be useful. This scatterplot, for example, shows that Pete Rose is an outlier, and his data should probably be removed to generate a better model:



All told, the SAS User Guide shows 17 different PROC REG output designs for this baseball example alone, including tables, graphs, and scatter plots, that would help our baseball-loving student refine a predictive model using this data set.[11]

B. The input formats and output designs of the SAS System are original expressions that embody many creative choices.

The central copyright question in this case is whether a competitor of SAS can shortcut the software design process by simply copying the input formats and output designs from the SAS System—with the intent of marketing a competing program that will look and feel the same to users. Copyright at its core protects original, creative expression. And Congress recognizes that copyright serves this critical function for software programs as it does for creative works in other media. There can be no serious dispute that a creative computer software system like the SAS System is entitled to copyright protection. *See Google LLC v. Oracle Am., Inc.*, 141 S. Ct. 1183, 1196 (2021) ("In 1980, Congress expanded the reach of the Copyright Act to include computer programs."). Multiple provisions of the Copyright Act recognize that a computer program may be protected by copyright. *See* 17 U. S. C. §§ 101, 109(b), 110(11), 117, 121(b)(2), 506(a). Consistent with these provisions, courts have long recognized that computer programs qualify for copyright protection

[11] *Id.*

as "original works of authorship fixed in any tangible medium of expression." 17 U.S.C. § 102(a); *see, e.g., See Computer Management Assistance Co. v. Robert F. DeCastro, Inc.*, 220 F.3d 396, 400 (5th Cir. 2000); *Atari Games Corp. v. Nintendo of Am., Inc.*, 975 F.2d 832, 838 (Fed.Cir.1992) ("As literary works, copyright protection extends to computer programs.").

Input formats easily fit within the Copyright Act's definition of a "computer program, which is "a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result." 17 U.S.C. § 101. The input formats associated with PROCs are just that: a set of instructions designed by SAS to generate certain results from the SAS System. Moreover, the input formats and output designs of a program like the SAS System are creative and expressive. Their design incorporates statistical analysis, to be sure, but their organization, language, and appearance are not dictated by math or by the coding process. As explained below, software programmers and designers make choices throughout the design process, to create a program that is intuitive, elegant, efficient, and useful.

An analogy to a much simpler format may be useful. Countless print and online texts teach students linear equations in the form of y=mx + b. Those texts depict straight lines on the coordinate x and y axes and use word problems to teach the concepts. The mathematical ideas and formulas cannot be copyrighted. But an

instructional math text that conveys original content, through its layout, graphical design, explanations, and word problems can certainly claim copyright protection.[12]

Just so here. Input formats and output designs are expressive elements in the medium of software. That is true for the SAS System and for countless other types of proprietary software. Educational assessment tools; medical diagnostic software; fitness-tracking programs; computer-aided design (CAD) programs—these are all examples of programs that organize and analyze information and generate results for users. Software developers *design* the format of those results. They decide whether to use text or graphics, tables or charts; pick fonts, colors, and column sizes; arrange information in a particular way; and choose labels and keywords. They decide on defaults and provide options where users can change the defaults. Their choices are entitled to copyright protection.

[12] *Cf., e.g.*, *Addison-Wesley Pub. Co. v. Brown*, 223 F. Supp. 219 (E.D.N.Y. 1963) (finding infringement of physics textbook); *Pearson Educ., Inc. v. Ishayev*, 9 F. Supp. 3d 328, 333 (S.D.N.Y. 2014) (solutions manuals for accounting textbooks protected by copyright).

1. The naming, selection, structure, and arrangement of input formats such as PROC statements reflect creative choices and decisions.

Input formats like PROCs reflect the creative choices of SAS. Programmers must decide which PROCS to have, how to express their input and output, and how to structure and arrange them.

To continue with PROC REG, its structure and syntax is described and explained in the SAS User Manual[13]:

- **PROC REG** <*options*>;
  <*label***:> MODEL** *dependents* = <*regressors*> </ *options*>;
  BY *variables*;
  FREQ *variable*;
  ID *variables*;
  VAR *variables*;
  WEIGHT *variable*;
  ADD *variables*;
  CODE <*options*>;
  DELETE *variables*;
  <*label***:> MTEST** <*equation*, *…*, *equation*> </ *options*>;
  OUTPUT <OUT=*SAS-data-set*> <*keyword=names*> <*…keyword=names*>;
  PAINT <*condition* |ALLOBS> </ *options*> |<STATUS |UNDO>;
  PLOT <*yvariable*xvariable*> <=*symbol*> <*…yvariable*xvariable*> <=*symbol*> </ *options*>;
  PRINT <*options*> <ANOVA> <MODELDATA>;
  REFIT ;
  RESTRICT *equation*, *…*, *equation*;
  REWEIGHT <*condition* |ALLOBS> </ *options*> |<STATUS |UNDO>;
  STORE <*options*>;
  <*label***:> TEST** *equation*, <, *…*, *equation*> </ *option*>;

[13] *See Syntax: REG Procedure*, SAS/STAT User's Guide (December 13, 2019), https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_reg_syntax.htm.

Software designers named PROC REG, chose and named each of the options listed above, developed a syntax for how the user interacts with PROC REG, and established default parameters. Each of the statements within PROC REG has a meaning; using a particular statement affects the analysis and output, for example by adding or deleting a variable or directing a certain kind of output, like a scatterplot. As another example, SAS has designed PROC REG to give the user the option to fit a model to the data; that's an option delineated by the statement MODEL.[14] The MODEL statement, in turn, is designed to permit dozens of further options.[15]

Another powerful SAS PROC for statistical analysis is PROC MIXED, a flexible statistical tool for fitting linear models to data. WPL copied PROC MIXED into its program. Here is the structure of PROC MIXED as shown in the SAS Manual:

[14] *See PROC REG Statement*, SAS/STAT User's Guide (December 13, 2019), https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_reg_syntax01.htm.

[15] *See MODEL Statement*, SAS/STAT User's Guide (December 13, 2019), https://documentation.sas.com/doc/en/pgmsascdc/9.4_3.4/statug/statug_reg_syntax08.htm.

SAS/STAT 14.1 User's Guide: The MIXED Procedure

Product: SAS/STAT
Procedure: Proc Mixed

PROC MIXED < *options* > ;
BY *variables* ;
CLASS *variable* < (**REF=** *option*) > . . . < *variable* < (**REF=** *option*) > > < / *global-option*
CODE < *options* > ;
ID *variables* ;
MODEL *dependent* = < *fixed-effects* > < / *options* > ;
RANDOM *random-effects* < / *options* > ;
REPEATED < *repeated-effect* > < / *options* > ;
PARMS (*value-list*) . . . < / *options* > ;
PRIOR < *distribution* > < / *options* > ;
CONTRAST *'label'* < *fixed-effect values* . . . >
< | *random-effect values* . . . >, . . . < / *options* > ;
ESTIMATE *'label'* < *fixed-effect values* . . . >
< | *random-effect values* . . . > < / *options* > ;
LSMEANS *fixed-effects* < / *options* > ;
LSMESTIMATE *model-effect lsmestimate-specification* < / *options* > ;
SLICE *model-effect* < / *options* > ;
STORE < **OUT=** > *item-store-name* < / **LABEL=** *'label'* > ;
WEIGHT *variable* ;

Again, as with PROC REG, the statements included in PROC MIXED have particular meaning and provide certain options to the user. SAS chose and named these statements, and created a particular structure, sequence, and grammar for the PROC.

There was no single, required way to author PROC REG or PROC MIXED or, more generally, to design an input format for these and other statistical analyses. Programmers could have done this work in countless different ways. As with any

organizational structure, a different system could be designed that, for example, grouped and separated tasks in different ways, used different names for the inputs, or ordered inputs in a different way. When designing input formats, programmers are making the kinds of choices typically recognized as creative.

The same is true of every PROC at issue. The SAS system, for example, contains PROCs for performing statistical analysis of variance, including ANOVA and GLM.[16] WPL elected to copy the inputs (and outputs) of both the ANOVA and GLM procedures, but of course it did not have to do so.[17] There are many other ways to design input formats for performing statistical analysis of variance, as offerings from other companies such as Microstrategy and SAP, which wrote their own programs without copying SAS's PROCs, amply demonstrate.[18]

---

[16] *See SAS/STAT 13.1 User's Guide The ANOVA Procedure* (December 2013), https://support.sas.com/documentation/onlinedoc/stat/131/anova.pdf at page 946 (page 6/67 of the PDF) ("The ANOVA procedure is one of several procedures available in SAS/STAT software for analysis of variance. The ANOVA procedure is designed to handle balanced data (that is, data with equal numbers of observations for every combination of the classification factors), whereas the GLM procedure can analyze both balanced and unbalanced data.").

[17] *See* E.D. Tex. Opening Br. at 23-24.

[18] *See, e.g., Var (variance of a sample)*, MicroStrategy Functions (April 2017), https://doc-archives.microstrategy.com/producthelp/10.8/FunctionsRef/Content/FuncRef/Var__variance_of_a_sample_.htm; Naveen Kumar, *SAP Analytics Cloud – All about Variance Analysis*, Visual BI Blogs (Feb. 21, 2020), https://visualbi.com/blogs/sap/sap-analytics-cloud/sap-analytics-cloud-variance-analysis/.

Indeed, the record in this case includes evidence that SAS programmers have a "wide range of creative design choices" for "statements and option names, syntax, and default parameters." E.D. Tex. Opening Br. at 20. They want to make the PROC concise and simple, user friendly and easy to understand. They try to choose names for the PROC and its inputs and options that are descriptive, intuitive, and easy to learn and remember. They may consider the likely intended user, including the user's training and education, and whether the likely user is in academics, industry, or government. Programmers consider whether terms are too short or too long; using abbreviations or not; the types of punctuation to use; whether the language expresses the underlying idea; whether the language is aesthetically pleasing or even witty; and the choice of default values. *See id.*; *see also SAS Institute*, 496 F. Supp. 3d (E.D. Tex. 2020), ECF Doc. 462-1, at 16 (hereafter E.D. Tex. Slides).

This evidence aligns with our experience as programmers and researchers. Affording copyright protection to input formats like PROCs is not akin to copyrighting an idea or a mathematical formula. Designing an input format like a PROC is a complex, creative process aimed at providing a user-friendly, expressive input format; it is not a mechanical exercise dictated by an underlying formula. SAS has invested in its design. Other data analysis programs came up with their own syntax and design. The record includes examples of user inputs for a regression

analysis of baseball statistics for SAS and two other programs. Here is the SAS input, using PROC REG:

```
ods select ParameterEstimates;
proc reg data=baseball;
id name team
league;
model logSalary = no_hits no_runs no_rbi no_bb yr_major cr_hits;
run;
```

E.D. Tex. Slides, at 17.

The open-source program R uses this input format to express the same problem:

```
reg_results<lm(logSalary~no_hits+no_runs+no_rbi+no_bb+yr_major+cr_hits)
```

E.D. Tex. Slides, at 18.

IBM's SPSS platform uses this syntax:

```
Regression
/dependent logSalary
/method=enter no_hits no_runs no_rbi no_bb yr_major cr_hits
```

E.D. Tex. Slides, at 19.

Someone starting from scratch to design a new data analysis program could readily come up with yet another way of accomplishing the same thing—possibly a better way. SAS's PROCs are not a data-analytics equivalent of the QWERTY keyboard. Indeed, even within the SAS System, users are not limited to the PROCs

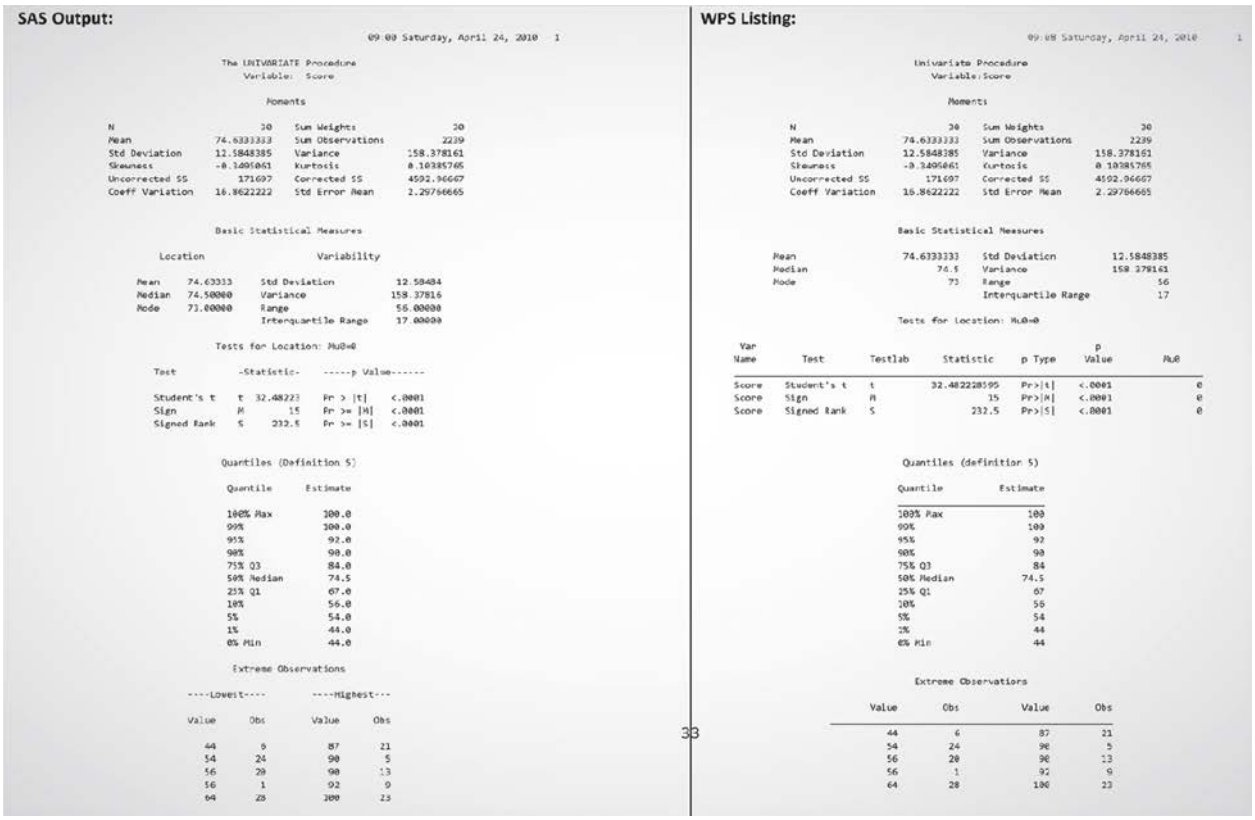
designed by SAS; they can write their own input formats.[19] Allowing a competitor to shortcut the development process and copy the work that SAS has invested in not only undermines the value of SAS's investment, it also diminishes incentives to innovate and improve.

2. The design and appearance of output designs like tables and graphs are chosen from virtually limitless options to be useful, appealing, and intuitive for the user.

The same is not only true for output designs—it is, if anything, more true. For a data management and analysis program like the SAS System, the options for organizing, displaying, and presenting results are virtually limitless. Text, tables, charts, and graphs are all options, and each of those formats can be designed in countless different ways. Font and color; text size; the ordering of information; overall and relative sizes for columns, rows, lines; choice and appearance of labels—all of these choices and more are reflected in output designs.

With a world of choices available for the design its outputs, WPL chose to duplicate the work of SAS. Here is a side-by-side comparison of the PROC UNIVARIATE output discussed above, with SAS on the left and WPL on the right:

[19] *See SAS/Toolkit*, https://support.sas.com/en/software/sas-toolkit-support.html (last visited May 20, 2021).



E.D. Tex. Slides, at 33.

Nothing about the underlying analysis or data requires this particular format. Nothing inherent in programming the code requires it either. The SPSS version looks entirely different:

Descriptive Statistics

	N	Range	Minimum	Maximum	Sum	Mean		Std. Deviation	Variance	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
Score	30	56.00	44.00	100.00	2239.00	74.6333	2.29767	12.58484	158.378	-.350	.427	.104	.833
Valid N (listwise)	30												

E.D. Tex. Slides, at 23.

And the output from R, again for the same problem, is in yet a different form:

output_Scores08062015.txt

```
> # https://support.sas.com/documentation/cdl/en/procstat/63104/HTML/default/viewer.htm#procstat_univariate_sect026.htm
> setwd("U:/Others/smalljobs")
>
> scores = scan("Scores.dat")
Read 30 items
>
> ###############################################################################
> ################ Results produced by the default function ###############
> ###############################################################################
>
> summary(scores)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  44.00   67.50   74.50   74.63   84.00  100.00
```

E.D. Tex. Slides, at 24.

This comparison simply confirms what the district court recognized: that instead of doing its own original work to design the output format, WPL set out to clone SAS.

Copyright law should protect the particular design choices made by SAS and other designers of proprietary programs. Granted, these are not great works of modern literature or critically acclaimed films. But the bar for copyright protection is not nearly so high. Software designers choose the organization and appearance of output formats with the goal of making the output useful, intuitive, visually appealing, and user-friendly. Whether ordering inventory for a store or tracking results of a vaccine trial, users *work* with these analyses. A business like SAS is in the business of making their work more efficient and effective. The choices made in output design is thus part of the value of SAS's product.

C. Because modernizing and improving software is also a creative process, the existence of an old public-domain version of a program does not excuse copying of a much different, current version.

Our interest, however, is not limited to SAS's ability to protect its investment. A sensible copyright regime encourages competitors not to copy, but to improve. When a competitor can merely duplicate the designs of an existing product, there is no incentive to make a better version of the software. As explained further below, the market and the public are better served by true competition, where a business that wants to compete for SAS's customers has to offer a better product.

One fact that the district court relied on is that a rudimentary version of SAS from the 1970 is in the public domain. We do not dispute that a competitor is free to copy expression that is in the public domain. But we caution that a 50-year-old version of a data analytics program bears little resemblance to a contemporary suite of business software in regular use by Fortune 500 companies. To stay competitive and relevant, a business like SAS has to continually modernize and improve its software. And that process of modernizing and rewriting the program reflects the same kind of creative process as the initial design. Filmmakers and authors continue to revise and retell classic stores like *Little Women* and *Pride and Prejudice*. Those new versions partake of material that is in the public domain but the new works as a whole are original, creative, and copyrightable. The same is true for a complex computer program like the SAS System.

Competitors are also free to use elements that are *scènes à faire*—but that doctrine is inapplicable here. The software in question is a product of SAS Institute's creative expression. Although WPL has asserted that the input formats are *scènes à faire*, it points to no evidence that extrinsic considerations such as mechanical specifications of the computers on which the SAS System runs or widely accepted standard programming practices dictated or circumscribed any of SAS Institute's creative choices. WPL has not (and cannot), for example, point to any elements of SAS's PROCs that are widely used by other programs, let alone programs that pre-date the work in question. They are not common or stock, and weren't when SAS Institute developed them. And even if WPL could do so for select aspects of a given PROC in isolation, that would not remotely begin to make the entire work, with its unique selection, structure, and arrangement of inputs, a "scene that must be done."

II. Robust copyright protection for creative, proprietary software like the SAS System is necessary to promote innovation and protect the economic investment in time-consuming, difficult work.

As specialists in the field, we know that creative, proprietary software like the SAS System is expensive to develop and maintain. The record here shows that the SAS System required "millions of man-hours of creative and difficult development and programming work on the part of thousands of SAS statisticians and programmers over the course of many years." E.D. Tex. Opening Br. at 18. Upfront development costs for a new product are substantial and require investors willing to

take a risk for (uncertain) long-term gains. And companies that launch successful products must continue to invest, to improve, and to innovate.

Proprietary software companies like SAS thus invest substantial resources in developing, maintaining, and constantly improving their products. SAS itself reinvests about 25% of its revenues in research and development—over $700 million annually. SAS Opening Br. 10 (citing Appx3351:12-20). Its research and development team includes about 2700 people. *Id.* (citing Appx3350:24-Appx3351:2, Appx3351:9-11). SAS has been recognized as one of the most innovative tech companies in the country.[20]

Other leading companies also invest enormous resources in developing and updating innovative products. For example, software giants Salesforce and Adobe each spent nearly $2 *billion* on research and development in fiscal year 2019.[21]

This kind of investment drives the creativity and innovation that are critical to the information economy. But companies will only make these investments if they can fairly expect a return. Proprietary companies expect to charge customers for

[20] *See* Press Release, *SAS Honored as a Stevie Award Winner in 2019 American Business Awards* (Jun. 26, 2019), http://bit.ly/2HhDHSz.

[21] *See* Press Release, *Adobe Surpasses $11 Billion in Annual Revenue*, 4 (Dec. 12, 2019), https://s23.q4cdn.com/979560357/files/docevents/2019/12/1/AdobeQ419Earnings.pdf; *2019 Annual Report: Celebrating 20 Years of Salesforce*, 39 (2019), https://s23.q4cdn.com/574569502/files/doc_financials/2019/Salesforce-FY-2019-Annual-Report.pdf.

using their software, just as publishers expect to charge customers for buying new books. And copyright protection is an essential part of the market for proprietary software, just as it for novels, photographs, and recordings. If competitors are free to copy software input formats and output designs to essentially clone programs like the SAS System, that hurts the original creators and destroys the incentives to create those programs in the first place.

Any suggestion that a permissive approach toward copying software promotes competition is true only in the narrowest and most short-sighted sense. In any medium, allowing a competitor to copy another's work may benefit consumers in the short-term, as the copier (who took without paying) uses its unfair market advantage to offer a product more cheaply. Over the long term, however, copyright protection promotes competition, innovation, and economic growth, because it provides incentives for businesses to continue developing and creating new and better programs. The U.S. economy reaps enormous benefits from the proprietary software industry.[22] Over 15,000 American software publishing companies

[22] *See* Greg Ip, *If the Economy Booms, Thank Software,* Wall St. J. (May 29, 2019), https://tinyurl.com/y5ofk6le; BSA Foundation, *The Growing $1 Trillion Economic Impact of Software* (Sept. 2017), https://tinyurl.com/y77xjgke.

collectively earned nearly $270 billion in 2019.[23] The industry grew five percent annually between 2014 and 2019, a trend that is expected to continue through 2024.[24]

In addition to fueling economic growth, robust copyright protection benefits consumers, the industry, and the public interest because it pushes competitors to create *better* programs. A clone of the SAS System is just that: a copy, that tries to do the same thing in the same way. If WPL and other competitors can simply appropriate the work of SAS, their incentive to innovative is greatly reduced. And it is innovation that drives progress.

We see the benefits of innovation and investment in our classrooms, academic programs, and research. Talented students are drawn to this field because of the promise of challenging, interesting work. They don't study for years and earn graduate degrees because they are excited to make copycat programs that duplicate existing works. They see the value of what they can create.

The decision below so misapplies core principles of copyright protection that it threatens this creative drive and undermines incentives for innovation. Consider output designs, for example. As discussed above, a data analytics program can generate its results in an almost endless variety of formats. And some of those

[23] Dan Cook, *Software Publishing Industry in the US*, IbisWorld, 4 (July 2019), http://bit.ly/2SFKdI9.

[24] *Id.* at 7, 10.

formats are better than others: clearer, easier to understand, better organized, or more aesthetically pleasing. If SAS's output designs are afforded appropriate copyright protection, then a competitor will be forced to rethink its output formats and develop different designs—designs that may meet different consumer needs or improve upon older versions. SAS has over time and with substantial investment developed an excellent program that generates highly useful statistical outputs. But that's not to say no one could do it better. Enforcing copyright drives the market to better designs.

The Supreme Court's recent decision in *Google v. Oracle* does not change this analysis. *Google*, of course, focuses on fair use rather than copyrightability. The Supreme Court assumed that Java's application programming interface (API) packages were copyrightable and did not reverse this Court's copyrightability opinion. *See Google*, 141 S. Ct. at 1197; *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1353 (Fed. Cir. 2014). Moreover, there is a critical distinction between copying a limited amount of code for a truly transformative purpose and wholesale copying of a competitor's input formats and output designs to make it easier to sell a product to the same customers for the same purposes. *See* SAS Opening Br., ECF Doc. 13, at 24-28 (detailing evidence that WPL intended to clone SAS software to market its product as a direct substitute). WPL did not need to use the same input formats or output designs to achieve the same functionality. It did so, presumably, because it was cheaper to develop and easier to convince customers to switch. A

company that copies Nike sneakers or Balenciaga boots could say the same thing: our customers want a product that looks and feels exactly the same, so we copied the exact design. That is infringement, pure and simple. *See id.* at 1208 (noting that holding does not modify any prior case law regarding "knockoff" products).

The question of copyrightability in this case thus matters beyond the dispute between SAS and WPL. The proprietary software industry is an important engine for economic growth. It provides high-quality jobs and drives technological advances. Copyright law is part of the legal framework necessary for companies to make investments in proprietary programs. The district court's ruling in this case, if not reversed, could do substantial harm to this important sector of the American economy.

CONCLUSION

For the foregoing reasons, this Court should reverse.

Dated: May 21, 2021

Respectfully submitted,

/s/ Bridget Asay

Bridget Asay
STRIS & MAHER LLP
28 Elm Street, Second Floor
Montpelier, VT 05602
Telephone: (802) 858-4285
basay@stris.com

Elizabeth Rogers Brannen
STRIS & MAHER LLP
777 S. Figueroa Street, Suite 3850
Los Angeles, CA 90017
Telephone: (213) 995-6800
Facsimile: (213) 261-0299
ebrannen@stris.com

Counsel for Amici Curiae

CERTIFICATE OF COMPLIANCE

I certify that this document complies with the word limit of Fed. R. App. P. 27(d)(2)(A) because, excluding the parts of the document exempted by Fed. R. App. P. 32(f), it contains 6,023 words.

Dated: May 21, 2021 /s/ *Bridget Asay*
Bridget Asay

CERTIFICATE OF SERVICE

I certify that on May 21, 2021, I electronically filed the foregoing brief with the Clerk of the Court by using the appellate CM/ECF system. I certify that the participants in the case are registered CM/ECF users and that service will be accomplished by the appellate CM/ECF system.

Dated: May 21, 2021 /s/ *Bridget Asay*
Bridget Asay