Linear Regression Model of Experience Level on Salary in Tech Industry

Group Alderwood: Zhaotian Li, Yonghao Li, Hongtianxu Hua
March 1st, 2020

1. Abstract

This report is concerned with the potential effects of personal and professional experience on the average salary received by full-time employees within tech companies in the US. We used Stack Overflow Developer Survey Results (2019) to build a linear regression model to identify the most influential factors. We found out that having a graduate degree has the highest impact on salary, along with other strong predictors. The linear regression model we build could provide insights to seasoned developers, job-seekers, and employers, regarding potential salary redistribution, job satisfaction, and work environment.

2. Introduction

Founded in 2008, Stack Overflow is one of the largest online communities for students, coders, and software developers to learn, share knowledge, and build careers. People use Stack Overflow to ask coding questions, find solutions, help solve problems, develop new skills, and find job opportunities (2020, Stack Overflow). According to its official data, by Feb 2020 there are over 19 million questions, 29 million answers, 12 million users in total and 11 million visits every day (2020, Stack Exchange). With an increasing number of programmers or coders, the user data gathered by Stack Overflow is a good sample to analyze the demographic pattern as well as socio-economic phenomenon or correlations within this community. The publicly available datasets for this purpose is Stack Overflow's annual Developer Survey. Since 2010, the site fields a survey each year covering everything from developers' favorite technologies to their job preferences. The 2019 survey we use contains nearly 90,000 answers.

The discussion of compensation is a never-ending question among tech workers, and many of them wonder how much their experience contributes to the growth of salary. By building a regression model upon the dataset, we would like to demystify the relationship between the developer's experience and salary. We hope to identify important factors contributing to the amount of salary received by a developer.

3. Research Question

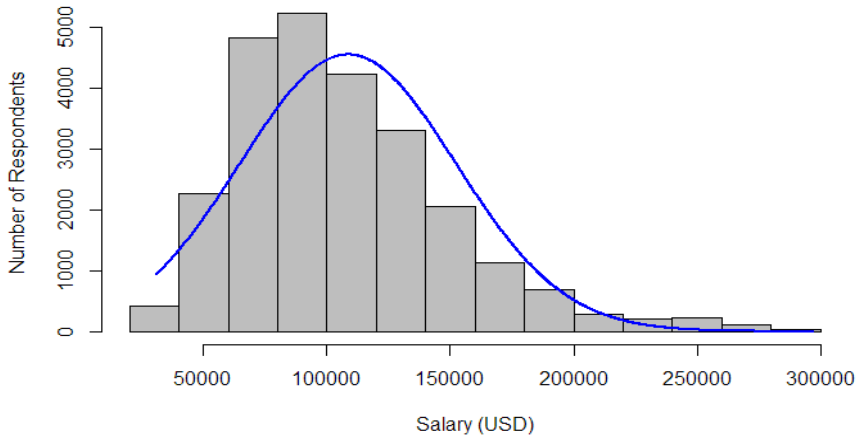
In this paper, we focus on the impact of developers' experience on their salary. The independent variables that are relevant to developers' experience include but not limited to: education level, the number of programming languages used, number of years coding professionally, etc.

4. Cleaning and exploring the salary data

We selected respondents who are full-time employed in the US to control the variations brought by the unbalanced economic factors worldwide. Although average salaries for developers differ by location, the dataset lacks this information. In an effort to keep as many observations as possible, this is the best approach to limiting the influence of geographic variance as well as currency. Extreme salary numbers could be reported by respondents from higher management. To account for this, we removed management job types such as Engineering manager, Product manager, Senior executive/VP, etc.

In Figure 1, the distribution histogram indicates positive skewness. Much fewer respondents claim to earn over $300,000 annually while most of them claim to make between $30,000 to $300,000 a year. In order to get an accurate model and to cover the majority of respondents, we arbitrarily removed observations with salary lower than $30,000 and higher than $300,000.

Figure 1: Normal Histogram of Salary Distribution



To fit some qualitative variables into our model, we created new columns summarizing the number of technical expertise (originally a multiple-choice question) that respondents worked with on a daily basis: “language_count”, "database_count " and "webframe_count". This is one way to represent a respondent’s work experience or coding capability; we will discuss the problem with this method in section 7.

5. Model

5.1 Motivation

The definition of “experience” on a tech worker could be broad. We not only selected directly relevant factors such as education level, but also try to infer the experience level of a respondent by checking their devotion to open source projects and the number of weekly working hours. We believe that these predictors could provide a solid foundation for analyzing the relationship between experience level and salary.

5.2 Assumptions / Prerequisites

Assumptions of linear regression model include linearity, homoscedasticity, positive variability in DV, normality, and there should not be any outliers or influential points affecting the model.

5.3 Analysis method

To discover the effect of selected variables on salary, we built a linear regression model to contain those variables. In order to normalize the model distribution, we decided to use the log scale of salary as the dependent variable. LR is a way of modeling the relationship between one or more variables and excels at interpreting the impact of a specific variable upon the outcome. By looking at the coefficient of a variable, we could explain how much the outcome changes with the change of 1 unit of the independent variable, by controlling all the other ones.

5.4 Linear Regression Models

Model 1 (shows influential points):

```
model_1 <- lm(log(Compensation_in_USD) ~ Education_Level + OpenSourcer +
Years_Coding_Professionally + Work_Hours_Per_Week + language_count + database_count +
webframe_count, data = sod_survey_selected[-which(abs(model1_dffits) > dffits_crit),])
```

Model 2 (updated model, adopted):

```
w <- abs(rstudent(model_1)) < 3 & abs(cooks.distance(model_1)) < 4/nrow(model_1$model)
model_2 <- update(model_1, weights=as.numeric(w))
```

Table 2: Summary Statistics of the Updated Model

```
Call:
lm(formula = log(Compensation_in_USD) ~ Education_Level + OpenSourcer +
    Years_Coding_Professionally + Work_Hours_Per_Week + language_count +
database_count + webframe_count, data = sod_survey_selected[-which(abs(model1_dffits)
> dffits_crit), ], weights = as.numeric(w))

                                  Estimate Std. Error t value  Pr(>|t|)
(Intercept)                     10.9387663  0.0163624 668.532 < 2e-16 ***
Education_LevelBachelor's degree  0.1261179  0.0050005  25.221 < 2e-16 ***
Education_LevelGraduate degree    0.2124113  0.0064454  32.955 < 2e-16 ***
OpenSourcerSometimes              0.0695256  0.0048091  14.457 < 2e-16 ***
OpenSourcerOften                  0.1244303  0.0048056  25.893 < 2e-16 ***
Years_Coding_Professionally       0.0204042  0.0002521  80.943 < 2e-16 ***
Work_Hours_Per_Week               0.0051520  0.0003620  14.234 < 2e-16 ***
language_count                   -0.0083948  0.0009644  -8.705 < 2e-16 ***
```

```
database_count                    0.0283213  0.0014007  20.219 < 2e-16 ***
webframe_count                   -0.0170829  0.0016609 -10.285 < 2e-16 ***
---
Residual standard error: 0.2918 on 22596 degrees of freedom
Multiple R-squared:  0.3098,     Adjusted R-squared:  0.3095
F-statistic:  1127 on 9 and 22596 DF,  p-value: < 2.2e-16
```

5.5 Results & Interpretation

From the model summary, we can see that the seven categories are significant predicting factors. The 3-star indicator at the end of each predictor means that these factors are statistically significant at alpha levels $\alpha = 0$ and $\alpha = 0.001$.

The list of coefficients (beta values) can be interpreted as the following:

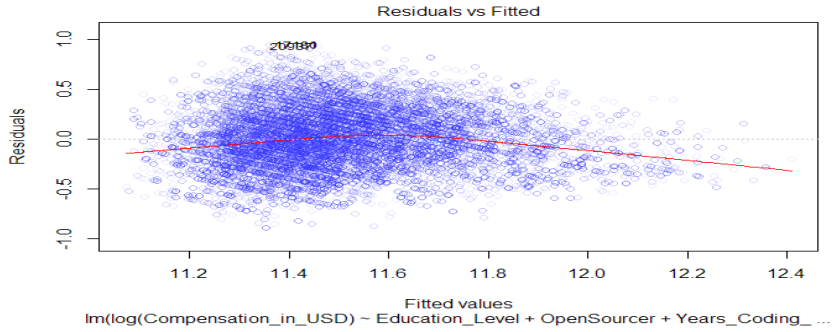
- Holding a Graduate Degree is predicted to lead to an increase in salary that is almost twice than that with a Bachelor's Degree;
- Employees who contribute or use open-source platforms often is predicted to have an increase in salary of more than twice than those who only use open-source occasionally;
- A year increase in coding professionally, lead to a 2.1% increase in salary since exp(0.0204042) = 1.0206;
- The negative coefficients of programming language count and web-frame count were surprising at first, but they might indicate that with an increase in languages used at work, typically a developer's expertise in one language might not be enough for high-level projects (which could lead to higher salary).
- Knowledge of database environments is important.

The adjusted R-squared score of 0.3095 means that these predictors explain about 31% of the variance within the dependent variable, which is the log scale of salary. Although this does not seem high at first glance, it is actually a very helpful statistic. About 31% of the variance in salary can be explained by these seven predictors alone, which is high considering the numerous other factors contributing to the distribution of salary (such as the cost of living in different cities, labor unions, company productivity, and market stability, etc.). These other factors were not available in the dataset. Besides, this model has a very small p-value that is less than 0.05, which indicates significance.

5.6 Statistical Uncertainty, Tests & Assumptions

a. Check linearity and homoscedasticity: Although there are small portions of residuals trailing off at the beginning and end, the red line is approximately flat at zero, suggesting a linear relationship between predictors and the log scale of salary. This plot also indicates that the variance is roughly constant since the mean of residual points is approximately zero.

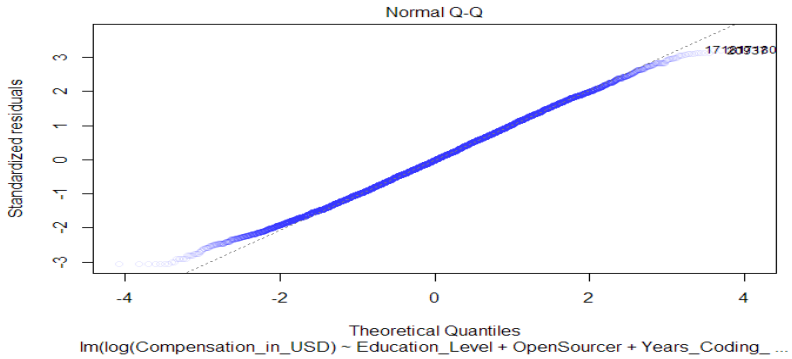
Figure 2: Residuals versus Fitted plot



b. Positive Variability in DV: var(sod_survey_selected$Compensation_in_USD) > 0
[1] TRUE

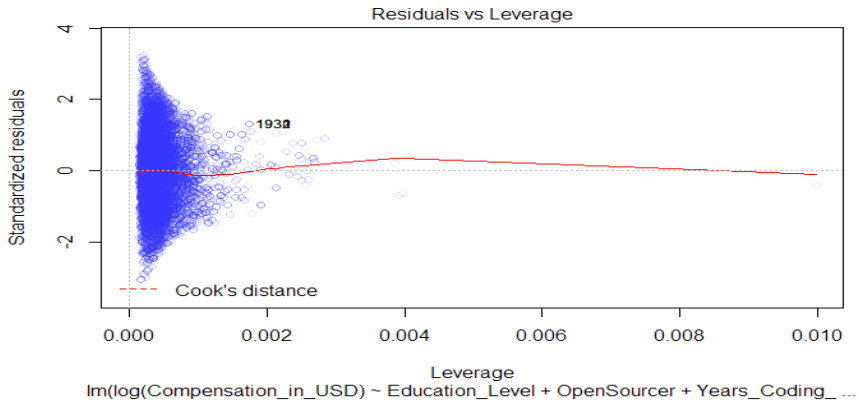
c. Check Normality: Normal Quantile-Quantile plot shows that the standardized residual points closely follow the reference line. This suggests that the model is distributed in a normal way.

Figure 3: Normal Quantile-Quantile plot



d. Check for outliers and influential points: Residuals vs. Leverage plot indicates that there are no outliers exceeding Cook's distance, and no significant influential points either.

Figure 4: Residuals vs. Leverage plot



5.7 Weaknesses of the model approach

Linear regression is sensitive to outliers. Although addressed in section 4, the cleaning process sacrifices the generalization and interpretability of the model. This approach oversimplifies real-world relationships since covariates and response variables rarely exhibit a linear relationship. Using multiple predictors also causes problems with model assumptions. Our preprocessing methods might also need improving because the initial model does not satisfy the assumptions perfectly. Using linear regression alone might not be the best way to deal with the variance and distribution of data within this huge dataset.

6. Ethical Issues

The dataset used in this study might be subject to privacy issue: the survey respondents might be concerned with privacy issues, although they must have been notified that the survey is anonymous and the result will be made public. No actual names or addresses were retained within the dataset. They might be comfortable filling out the survey, but there's no guarantee that they will be honest when answering questions related to private matters. This study does not promote stereotypes or discrimination against or even among developers. We believe that it is everyone’s free choice to decide what languages to learn and use, what front-end and back-end technologies to adopt, when to let children start learning computer languages, etc. Any correlations revealed by this study are bias-free.

7. Shortcomings and Weaknesses

The design of the survey might bias survey respondents' answers toward certain questions. Other influential variables are missing from the survey, such as the geographic location of the workplace, the industry they work for, along with other factors. If we have access to additional information, we might be able to build a better model. In terms of the dependent variable, there might be better measures of work achievement other than the amount of salary received. For example, using quantitative metrics to describe a developer's contribution of coding within the company might be another way to go.

In the original survey dataset, some columns contain multiple-choice questions or free responses. For data cleaning purposes, we converted a few columns into numeric values by counting the options that appeared in one answer (such as “language_count”). We did not manage to process these variables in a more sensible way, thus unable to explore further into their possible impact on salary.

Appendix 1 - Request for own dataset

We gained approval to use our own dataset in the problem set. Email information below:

Email：george.hua@mail.utoronto.ca

Date: 18 Feb, 2020

Name: Hongtianxu Hua (George)

Appendix 2 - Variable Reference Table

Independent Variables Name	Explanation	Type	Value
Education_Level		Categorical	“Less than bachelor”, “Bachelor’s degree”, “Graduate degree”
OpenSourcer	How much the responder participate in open source projects	Categorical	“Never”, “Sometimes”, “Often”
Years_Coding_ Professionally		Numeric	Number
Work_Hours_Per_Week		Numeric	Number
language_count	How many programming languages the responder works with	Numeric	Number
database_count	Number of database environments work with	Numeric	Number
webframe_count	Number of web frameworks work with	Numeric	Number

Appendix 3: Codes

```
# Load in necessary libraries
library(tidyverse)
library(stringr)
library(skimr)
library(janitor)
library(readr)
library(e1071)
library(data.table)
library(tidyr)
library(rcompanion)
library(MASS)
library(scales)


# First download and read the dataset from Kaggle.com
sod_survey <- read_csv("https://www.kaggle.com/mchirico/
                       stack-overflow-developer-survey-results-2019/")
# sod_survey <- read_csv("./survey_results_public.csv")

# Use filter to select data portion that fits our research scope
# (Using Julia Silge's codes on website as reference)
sod_survey <- sod_survey %>%
  filter(
    Country == "United States",
    Employment == "Employed full-time",
    ## Remove atypical salary ranges
    ConvertedComp > 3e4,
    ConvertedComp < 3e5
  )

sod_survey %>% summarise_all(.funs = funs(sum(is.na(.))))

atypical_employment <- sod_survey %>%
  filter(str_detect(DevType, "Engineering manager|Product manager|
                    Senior executive/VP|Academic researcher
                    |Scientist|Educator"))
# This step only removes management job types as independent variables
# but does not guarantee the eliminateion of outliers.
# Outliers can be caused by intentional typos or jokes within responses.
head(sod_survey$LanguageWorkedWith)

# Pre-processing:
# Create new columns containing the numbers of programming languages,
# database, and webframe used by respondents on a daily basis
sod_survey$language_count <- sapply(sod_survey$LanguageWorkedWith,
                                    function(x) lengths(strsplit(x, split = ";")))
```

```r
sod_survey$database_count <- sapply(sod_survey$DatabaseWorkedWith,
                                    function(x) lengths(strsplit(x, split = ";")))

sod_survey$webframe_count <- sapply(sod_survey$WebFrameWorkedWith,
                                    function(x) lengths(strsplit(x, split = ";")))


# Retain and Optimize the columns that are potentially impactful to salary
# (Using Julia Silge's codes on website as reference)
sod_survey_selected <- sod_survey %>%
  anti_join(atypical_employment) %>%
  transmute(Respondent,
    EdLevel = fct_collapse(EdLevel,
      `Less than bachelor's` = c(
        "I never completed any formal education",
        "Primary/elementary school",
        "Secondary school (e.g. American high school,
        German Realschule or Gymnasium, etc.)",
        "Some college/university study without earning a degree",
        "Associate degree"
      ),
      `Bachelor's degree` = "Bachelor’s degree (BA, BS, B.Eng., etc.)",
      `Graduate degree` = c(
        "Other doctoral degree (Ph.D, Ed.D., etc.)",
        "Master’s degree (MA, MS, M.Eng., MBA, etc.)",
        "Professional degree (JD, MD, etc.)"
      )
    ),
    DevType,
    Age1stCode,
    OpenSourcer = fct_collapse(OpenSourcer,
      Never = "Never",
      Sometimes = "Less than once per year",
      Often = c(
        "Less than once a month but more than once per year",
        "Once a month or more often"
      )
    ),
    OpenSourcer = fct_rev(OpenSourcer),
    YearsCodePro = parse_number(YearsCodePro),
    Gender = case_when(
      str_detect(Gender, "Non-binary") ~ "Non-binary",
      TRUE ~ Gender
    ),
    CareerSat,
    JobSat,
    ConvertedComp,
    WorkLoc,
    WorkWeekHrs,
    language_count,
    database_count,
    webframe_count
  )
```

```r
# Change sub-catgories of DevType to be more readable
# (Using Julia Silge's codes on website as reference)
sod_survey_selected <- sod_survey_selected %>%
  mutate(DevType = str_split(DevType, pattern = ";")) %>%
  unnest(DevType) %>%
  mutate(
    DevType = case_when(
      str_detect(str_to_lower(DevType), "data scientist") ~ "Data scientist",
      str_detect(str_to_lower(DevType), "data or business") ~ "Data analyst",
      str_detect(str_to_lower(DevType), "desktop") ~ "Desktop",
      str_detect(str_to_lower(DevType), "embedded") ~ "Embedded",
      str_detect(str_to_lower(DevType), "devops") ~ "DevOps",
      str_detect(DevType, "Engineer, data") ~ "Data engineer",
      str_detect(str_to_lower(DevType), "site reliability") ~ "DevOps",
      TRUE ~ DevType
    ),
    DevType = str_remove_all(DevType, "Developer, "),
    DevType = str_to_sentence(DevType),
    DevType = str_replace_all(DevType, "Qa", "QA"),
    DevType = str_replace_all(DevType, "Sre", "SRE"),
    DevType = str_replace_all(DevType, "Devops", "DevOps")
  ) %>%
  filter(!is.na(DevType)) %>%
  filter(Gender %in% c("Man", "Woman"))

# Rename column headers to be comprehensible
sod_survey_selected <- sod_survey_selected[, -1] %>%
  rename(Education_Level = EdLevel, Developer_Type = DevType, Age_First_Code = Age1stCode,
         Years_Coding_Professionally = YearsCodePro, Career_Satisfaction = CareerSat,
         Job_Satisfaction = JobSat, Compensation_in_USD = ConvertedComp,
         Work_Location = WorkLoc, Work_Hours_Per_Week = WorkWeekHrs)

colnames(sod_survey_selected)

# Remove missing values
sod_survey_selected <- sod_survey_selected %>%
  drop_na()

sod_survey_selected %>%
  summarise_all(.funs = funs(sum(is.na(.))))


# View historgram of ConvertedComp to get a general sense of salary distribution
# Heavily-skewed (Positive Skewness)
skewness(sod_survey_selected$Compensation_in_USD)

options(scipen=10000)
plotNormalHistogram(sod_survey_selected$Compensation_in_USD,
                    main = "Figure 1: Normal Histogram of Salary Distribution",
                    xlab="Salary (USD)",
                    ylab="Number of Respondents")

# Explore the respondents with min or max salary
```

```r
sod_survey_selected[which.max(sod_survey_selected$Compensation_in_USD), ]

sod_survey_selected[which.min(sod_survey_selected$Compensation_in_USD), ]


# Feature Selection:
# From the results below, we can see that variables such as "Age First Code"
# might not be one of the most influential elements.
# 1. Backward Elimination
full <- lm(Compensation_in_USD~.,data=sod_survey_selected)
stepB <- stepAIC(full, direction= "backward", trace=FALSE)
summary(stepB)

# 2. Forward Selection
full_2 <- lm(Compensation_in_USD~., data=sod_survey_selected)
null <- lm(Compensation_in_USD~1,data=sod_survey_selected)
stepF <- stepAIC(null, scope=list(lower=null, upper=full), direction= "forward",
                 trace=FALSE)
summary(stepF)

# Linear Regression:
# First attempt to remove influential points
p <- length(model_1$coefficients)
n <- nrow(model_1$model)
dffits_crit = 2 * sqrt((p + 1) / (n - p - 1))
model1_dffits <- dffits(model_1)

# Building model 1
model_1 <- lm(log(Compensation_in_USD) ~ Education_Level + OpenSourcer +
                Years_Coding_Professionally + Work_Hours_Per_Week + language_count +
                database_count + webframe_count,
              data = sod_survey_selected[-which(abs(model1_dffits) > dffits_crit),])
summary(model_1)

# Testing and Analysis on Model Assumptions:
# Testing linearity and constant variance
plot(model_1, 1, col=rgb(red=0.2, green=0.2, blue=1.0, alpha=0.1))
# Testing normality
plot(model_1, 2, col=rgb(red=0.2, green=0.2, blue=1.0, alpha=0.1))
# Shows no outliers but influential points
plot(model_1, 5, col=rgb(red=0.2, green=0.2, blue=1.0, alpha=0.1))

# Check for positive variability in salary as the dependent variable
var(sod_survey_selected$Compensation_in_USD) > 0

# Second attempt to remove influential points
w <- abs(rstudent(model_1)) < 3 & abs(cooks.distance(model_1)) < 4/nrow(model_1$model)

# Building model 2
model_2 <- update(model_1, weights=as.numeric(w))
summary(model_2)
# Testing linearity and constant variance
plot(model_2, 1, col=rgb(red=0.2, green=0.2, blue=1.0, alpha=0.1))
```

```r
# Testing normality
plot(model_2, 2, col=rgb(red=0.2, green=0.2, blue=1.0, alpha=0.1))
# Shows no outliers and no influential points
plot(model_2, 5, col=rgb(red=0.2, green=0.2, blue=1.0, alpha=0.1))
```

Appendix 4: References

Stack Overflow 2019 Survey Collection Methodology. https://insights.stackoverflow.com/survey/2019#methodology

Julia Silge (2019). Modeling Salary and Gender in The Tech Industry. https://juliasilge.com/blog/salary-gender/

Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686. https://doi.org/10.21105/joss.01686

Hadley Wickham (2019). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.4.0. https://CRAN.R-project.org/package=stringr

Elin Waring, Michael Quinn, Amelia McNamara, Eduardo Arino de la Rubia, Hao Zhu and Shannon Ellis (2019). Skimr: Compact and Flexible Summaries of Data. R package version 2.0.2. https://CRAN.R-project.org/package=skimr

Sam Firke (2019). janitor: Simple Tools for Examining and Cleaning Dirty Data. R package version 1.2.0. https://CRAN.R-project.org/package=janitor

Hadley Wickham, Jim Hester and Romain Francois (2018). readr: Read Rectangular Text Data. R package version 1.3.1. https://CRAN.R-project.org/package=readr

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel and Friedrich Leisch (2019). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-3. https://CRAN.R-project.org/package=e1071

Matt Dowle and Arun Srinivasan (2019). data.table: Extension of `data.frame`. R package version 1.12.8. https://CRAN.R-project.org/package=data.table

Hadley Wickham and Lionel Henry (2020). tidyr: Tidy Messy Data. R package version 1.0.2. https://CRAN.R-project.org/package=tidyr

Salvatore Mangiafico (2020). rcompanion: Functions to Support Extension Education Program Evaluation. R package version 2.3.25. https://CRAN.R-project.org/package=rcompanion

Venables, W. N. & Ripley, B. D. (2002) "MASS". Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0.

Hadley Wickham and Dana Seidel (2019). scales: Scale Functions for Visualization. R package version 1.1.0. https://CRAN.R-project.org/package=scales

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for StatisticalComputing, Vienna, Austria. http://www.R-project.org/.3

Stack Overflow Developer Survey Results 2019. Kaggle.com. https://www.kaggle.com/mchirico/stack-overflow-developer-survey-results-2019#survey_results_public.csv

Stack Exchange. https://stackexchange.com/sites?view=list#users