r2spss: Format R Output to Look Like SPSS

Andreas Alfons
Erasmus University Rotterdam

Abstract

The R package **r2spss** allows to create plots and LaTeX tables that look like SPSS output for use in teaching materials. Rather than copying-and-pasting SPSS output into documents, R code that mocks up SPSS output can be integrated directly into dynamic LaTeX documents with tools such as the R package **knitr**. Package **r2spss** provides functionality for statistical techniques that are typically covered in introductory statistics courses: descriptive statistics, common hypothesis tests, ANOVA, and linear regression, as well as box plots, histograms, scatter plots, and line plots (including profile plots).

Keywords: R, SPSS, statistics, teaching.

1. Introduction

Many academic programs in the social sciences or economics require to teach statistics with SPSS (IBM Corp. 2021). Preparing teaching materials in this case typically involves copying-and-pasting SPSS output into documents or slides, which is cumbersome and prone to errors. Moreover, this approach is not scalable for regular updates of the materials, or for individualizing assignments and exams in order to combat fraud. On the other hand, tools such as package **knitr** (Xie 2015, 2021) for integrating the statistical computing environment R (R Core Team 2021) and the document preparation system LaTeX (e.g., Mittelbach, Goossens, Braams, Carlisle, and Rowley 2004) make preparing teaching materials easier, less error-prone, and more scalable. There are even specialized tools such as package **exams** (Grün and Zeileis 2009; Zeileis, Umlauf, and Leisch 2014; Zeileis, Grün, Leisch, and Umlauf 2020) that allow assignments and exams to be individualized in a scalable manner. Package **r2spss** (Alfons 2021) makes it possible to leverage those developments for creating teaching materials with SPSS output by mocking up such output with R.

2. LaTeX documents containing output from r2spss

We first load the package to discuss its main functionality to generate LaTeX tables.

```
R> library("r2spss")
```

2.1. LaTeX requirements

LaTeX tables created with package **r2spss** build upon several LaTeX packages. A LaTeX style file that includes all requirements can be produced with function `r2spss.sty()`. By default, it prints the content of the style file on the R console, but its only argument `path` can be used to specify the path to a folder in which to put the file *r2spss.sty*. For instance, the following command can be used to put the style file in the current working directory.

```
R> r2spss.sty(path = ".")
```

After putting the style file in the folder that contains your LaTeX document, the following command should be included in the preamble of your LaTeX document, i.e., somewhere in between `\documentclass{}` and `\begin{document}`.

```
\usepackage{r2spss}
```

2.2. Workhorse functions to create LaTeX tables with r2spss

Functions in package **r2spss** create certain R objects, whose `print()` method prints the LaTeX tables that mimic the corresponding SPSS output. Essentially, such a `print()` method first calls function `to_SPSS()`, which produces an object of class `"SPSS_table"`. Its component `table` contains a data frame of the results in SPSS format. Other components of the object contain any necessary additional information of the SPSS table, such as the main title, the header layout, or footnotes. Afterwards, the `print()` method calls function `to_latex()` with the `"SPSS_table"` object to print the LaTeX table.

These two function can also be called separately by the user, which allows for further customization of the LaTeX tables. Some examples can be found in the help file of `to_SPSS()` or `to_latex()`, which can be accessed from the R console with `?to_SPSS` and `?to_latex`, respectively. In addition, the `"data.frame"` method of `to_latex()` allows to extend the functionality of **r2spss** with additional LaTeX tables that mimic the look of SPSS output.

Package **r2spss** can create output that mimics the look of current SPSS versions, as well as the look of older versions. The above mentioned functions contain the argument `version` for specifying which type of output to create. Possible values are `"modern"` to mimic recent versions and `"legacy"` to mimic older versions. LaTeX tables that mimic the look of recent SPSS version thereby build upon the LaTeX package **nicematrix** (Pantigny 2021) and its `NiceTabular` environment, which is preferred for its seamless display of background colors in the table.

However, **r2spss** requires **nicematrix** version 6.5 (2022-01-23) or later. It is also important to note that tables using the `NiceTabular` environment may require several LaTeX compilations to be displayed correctly. For portability reasons, this vignette therefore only displays LaTeX tables that mimic the simpler look of older SPSS versions. For convenience, such a global preference within an R session can be set with the accessor function `r2spss_options$set()`.

```
R> r2spss_options$set(version = "legacy")
```

2.3. Dynamic documents and knitr options

Package **r2spss** is the most useful when writing dynamic LaTeX documents with tools such as the R package **knitr** (Xie 2015, 2021). When creating LaTeX tables in R code chunks with **knitr**, the output of the chunk should be written directly into the output document by setting the chunk option `results='asis'`. For more information on **knitr** chunk options, in particular various options for figures, please see https://yihui.org/knitr/options/.

3. Illustrations: Using package r2spss

Several examples showcase the functionality of **r2spss** to mock up SPSS tables and graphics.

3.1. Example data sets

The following two data sets from package **r2spss** will be used to illustrate its functionality: `Eredivisie` and `Exams`. The former contains information on all football players in the Dutch Eredivisie, the highest men's football league in the Netherlands, who played at least one match in the 2013-14 season. The latter contains grades for an applied statistics course at Erasmus University Rotterdam for students who took both the regular exam and the resit.

```
R> data("Eredivisie")
R> data("Exams")
```

Among other information, the `Eredivisie` data contain the market values of the football players. In many examples, we will use the logarithm of the market values rather that the market values themselves, so we add those to the data set.

```
R> Eredivisie$logMarketValue <- log(Eredivisie$MarketValue)
```

3.2. Descriptive statistics and plots

Descriptive statistics can be produced with function `descriptives()`, for example of the age, minutes played, and logarithm of market value of football players in the `Eredivisie` data.

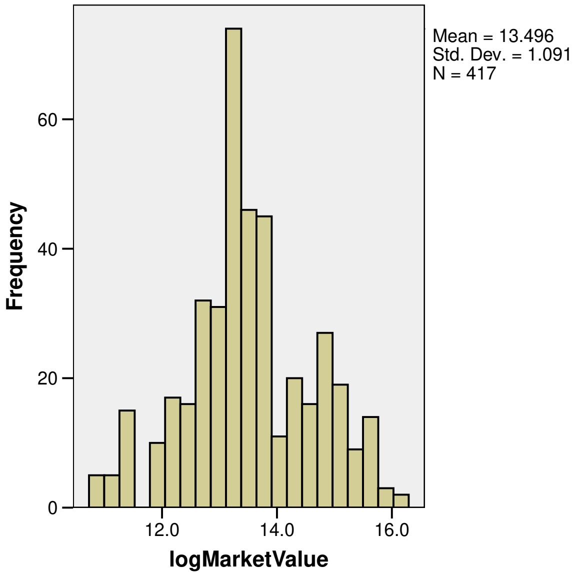
```
R> descriptives(Eredivisie, c("Age", "Minutes", "logMarketValue"))
```

Descriptive Statistics

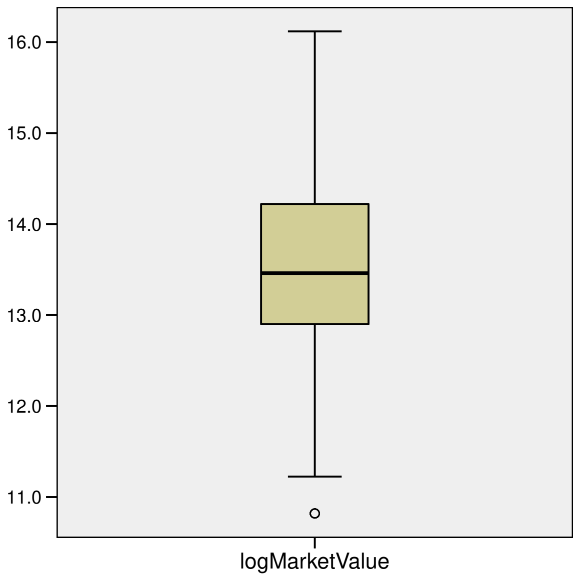
	N	Minimum	Maximum	Mean	Std. Deviation
Age	417	16	38	24.36	3.99
Minutes	417	1	3060	1425.81	972.08
logMarketValue	417	10.82	16.12	13.50	1.09
Valid N (listwise)	417				

Functions `histogram()` and `box_plot()` can be used to create a histogram or box plot, respectively, of a specified variable.

```
R> histogram(Eredivisie, "logMarketValue")
```

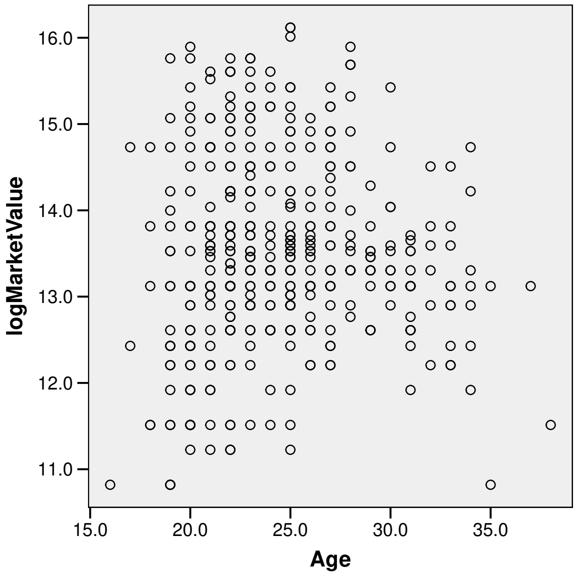



```
R> box_plot(Eredivisie, "logMarketValue")
```

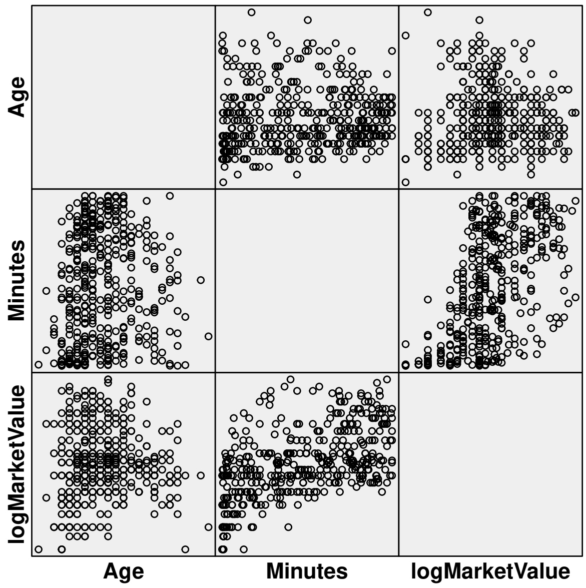


A scatter plot or scatter plot matrix can be produced with function `scatter_plot()` by specifying the corresponding variables.

```
R> scatter_plot(Eredivisie, c("Age", "logMarketValue"))
```




```
R> scatter_plot(Eredivisie, c("Age", "Minutes", "logMarketValue"))
```



3.3. Analyzing one sample

With the `Exams` data, we can perform a one-sample t test on whether the average grade on the resit exam differs from 5.5, which is the minimum passing grade in the Netherlands. For this purpose, we can use function `t_test()` with a single variable as well as the value under the null-hypothesis.

```
R> t_test(Exams, "Resit", mu = 5.5)
```

One-Sample Statistics

	N	Mean	Std. Deviation	Std. Error Mean
Resit	45	5.598	1.438	.214

One-Sample Test

	Test Value = 5.5					
					95% Confidence Interval of the Difference	
	t	df	Sig.(2-tailed)	Mean Difference	Lower	Upper
Resit	.456	44	.651	.098	-.334	.530

3.4. Analyzing paired observations

Similarly, we can perform a paired-sample t test on whether the average grades differ between the regular exam and the resit by supplying the two corresponding variables to function `t_test()`.

```
R> t_test(Exams, c("Resit", "Regular"))
```

Paired Samples Statistics

	N	Mean	Std. Deviation	Std. Error Mean
Resit	45	5.598	1.438	.214
Regular	45	3.971	1.142	.170

Paired Samples Test

	Paired Differences							
				95% Confidence Interval of the Difference				
	Mean	Std. Deviation	Std. Error Mean	Lower	Upper	t	df	Sig.(2-tailed)
Resit - Regular	1.627	1.434	.214	1.196	2.057	7.610	44	.000

As nonparametric alternatives, we can perform a Wilcoxon signed rank test with function `wilcoxon_test()` or a sign test with function `sign_test()`.

```
R> wilcoxon_test(Exams, c("Regular", "Resit"))
```

Ranks

		N	Mean Rank	Sum of Ranks
Resit - Regular	Negative Ranks	3[a]	14.00	42.00
	Positive Ranks	41[b]	23.12	948.00
	Ties	1[c]		
	Total	45		

a. Resit < Regular
b. Resit > Regular
c. Resit = Regular

Test Statistics[a]

	Resit - Regular
Z	-5.288[b]
Asymp. Sig. (2-tailed)	.000

a. Wilcoxon Signed Ranks Test
b. Based on positive ranks.

```
R> sign_test(Exams, c("Regular", "Resit"))
```

Frequencies

		N
Resit - Regular	Negative Differences[a]	3
	Positive Differences[b]	41
	Ties[c]	1
	Total	45

a. Resit < Regular
b. Resit > Regular
c. Resit = Regular

Test Statistics[a]

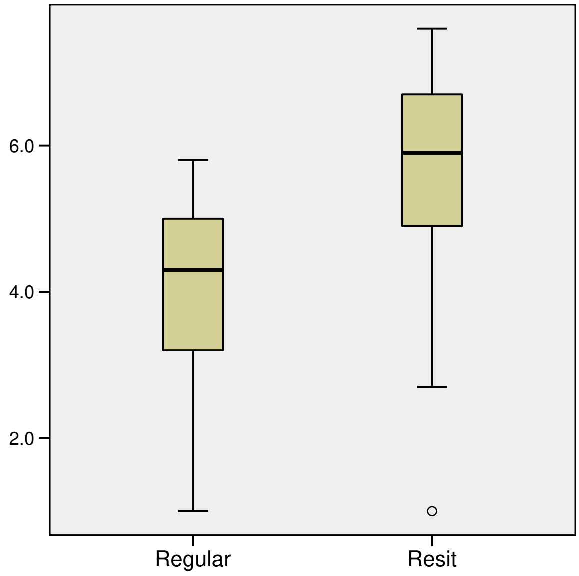
	Resit - Regular
Z	-5.578
Asymp. Sig. (2-tailed)	.000

a. Sign Test

Note that the order of the variables in the nonparametric test is reversed compared to the paired-sample t test, but all three tests compute the differences in the form `Resit - Regular`. This behavior is carried over from SPSS.

To check which of these tests are suitable for the given data, we can for example use a box plot. Function `box_plot()` allows to specify multiple variables to be plotted.

```
R> box_plot(Exams, c("Regular", "Resit"))
```



3.5. Comparing two groups

An independent-samples t test can be performed with function `t_test()` by specifying the numeric variable of interest as well as a grouping variable. As an example, we test whether the average log market values differ between Dutch and foreign football players.

```
R> t_test(Eredivisie, "logMarketValue", group = "Foreign")
```

Group Statistics

	Foreign	N	Mean	Std. Deviation	Std. Error Mean
logMarketValue	0	279	13.345	1.108	.066
	1	138	13.801	.994	.085

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
logMarketValue	Equal variances assumed	.979	.323	-4.085	415	.000	-.455	.111	-.675	-.236
	Equal variances not assumed			-4.237	301.040	.000	-.455	.107	-.667	-.244

As a nonparametric alternative, we can perform a Wilcoxon rank sum test with function `wilcoxon_test()` in a similar manner. Note that it is not necessary to use the logarithms of the market values here, as this test works with ranks instead of the observed values.

```
R> wilcoxon_test(Eredivisie, "MarketValue", group = "Foreign")
```

Ranks

		N	Mean Rank	Sum of Ranks
MarketValue	0	279	192.08	53590.00
	1	138	243.21	33563.00
	Total	417		

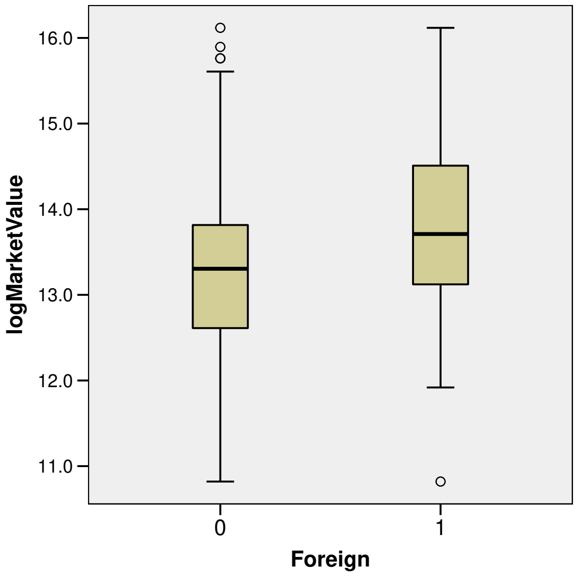
Test Statistics[a]

	MarketValue
Mann-Whitney U	14530.000
Wilcoxon W	53590.000
Z	-4.083
Asymp. Sig. (2-tailed)	.000

a. Grouping Variable: Foreign

We can again use a box plot to check whether the t test is suitable for the given data, as function `box_plot()` allows to specify a grouping variable as well.

```
R> box_plot(Eredivisie, "logMarketValue", group = "Foreign")
```



3.6. Comparing multiple groups

For comparing the means of multiple groups, one-way ANOVA can be performed with function `ANOVA()`. Here we test whether there are differences among the average log market values for players on different positions.

```
R> oneway <- ANOVA(Eredivisie, "logMarketValue", group = "Position")
R> oneway
```

Descriptives

logMarketValue

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean Lower Bound	95% Confidence Interval for Mean Upper Bound	Minumum	Maximum
Goalkeeper	35	13.343	1.322	.223	12.889	13.797	10.820	15.425
Defender	137	13.396	.986	.084	13.230	13.563	10.820	15.687
Midfielder	121	13.568	1.115	.101	13.367	13.769	10.820	16.118
Forward	124	13.580	1.108	.100	13.383	13.777	10.820	16.118
Total	417	13.496	1.091	.053	13.391	13.601	10.820	16.118

Test of Homogeneity of Variances

Levene Statistic	df1	df2	Sig.
2.666	3	413	.047

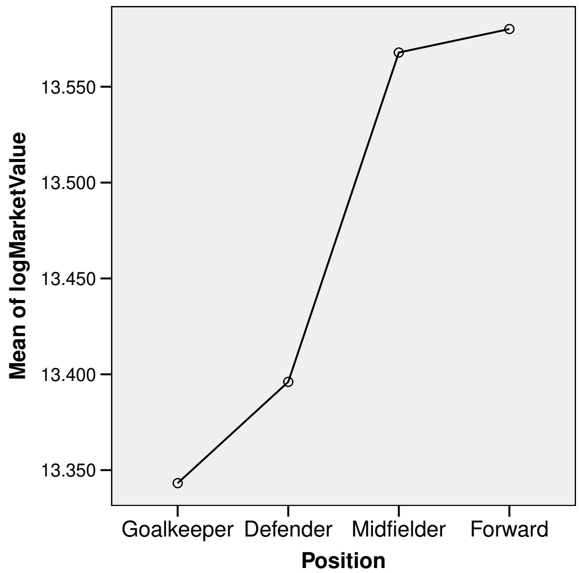
ANOVA

logMarketValue

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	3.687	3	1.229	1.032	.378
Within Groups	491.786	413	1.191		
Total	495.474	416			

The `plot()` method for the resulting object produces a profile plot.

```
R> plot(oneway)
```



A nonparametric alternative based on ranks is the Kruskal-Wallis test, which can be applied with function `kruskal_test()`. It is again not necessary to use the logarithms of the market values for this test.

```
R> kruskal_test(Eredivisie, "MarketValue", group = "Position")
```

Ranks

	Position	N	Mean Rank
MarketValue	Goalkeeper	35	196.01
	Defender	137	197.52
	Midfielder	121	217.17
	Forward	124	217.38
	Total	417	

Test Statistics[a,b]

	MarketValue
Chi-Square	2.814
df	3
Asymp. Sig.	.421

a. Kruskal Wallis Test
b. Grouping Variable: Position

Similarly, two-way ANOVA can be performed by supplying two grouping variables to function `ANOVA()`.

```
R> twoway <- ANOVA(Eredivisie, "logMarketValue",
+                  group = c("Position", "Foreign"))
R> twoway
```

Descriptive Statistics

Dependent Variable: logMarketValue

Position	Foreign	Mean	Std. Deviation	N
Goalkeeper	0	13.254	1.465	24
	1	13.538	.972	11
	Total	13.343	1.322	35
Defender	0	13.289	1.033	99
	1	13.675	.795	38
	Total	13.396	.986	137
Midfielder	0	13.474	1.160	84
	1	13.781	.987	37
	Total	13.568	1.115	121
Forward	0	13.304	1.016	72
	1	13.963	1.126	52
	Total	13.580	1.108	124
Total	0	13.345	1.108	279
	1	13.801	.994	138
	Total	13.496	1.091	417

Levene's Test of Equality of Error Variances[a]

Dependent Variable: logMarketValue

Levene Statistic	df1	df2	Sig.
2.658	7	409	.011

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.
a. Design: Intercept + Position + Foreign + Position * Foreign

Tests of Between-Subject Effects

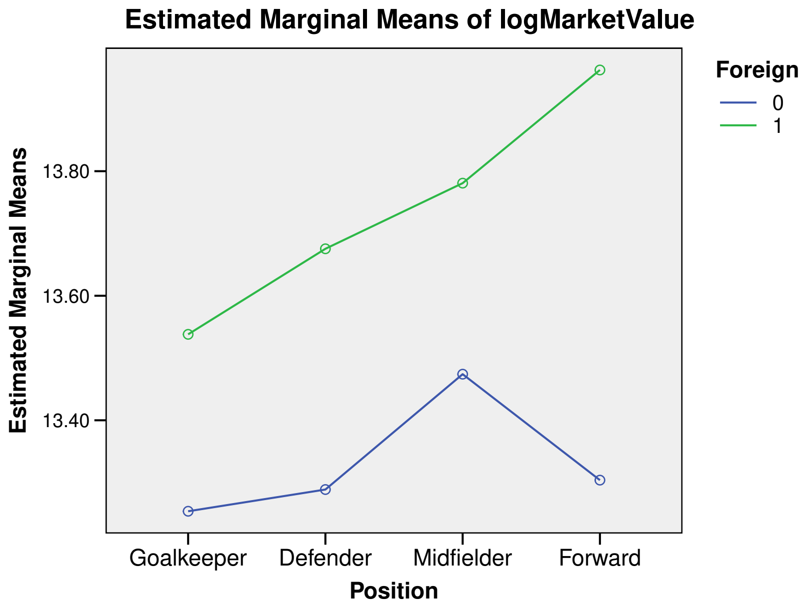
Dependent Variable: logMarketValue

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	23.914[a]	7	3.416	2.963	.005
Intercept	48638.419	1	48638.419	42185.739	.000
Position	2.578	3	.859	.745	.525
Foreign	11.104	1	11.104	9.631	.002
Position * Foreign	2.158	3	.719	.624	.600
Error	471.560	409	1.153		
Total	49133.893	417			
Corrected Total	495.474	416			

a. R Squared = .048 (Adjusted R Squared = .032)

We can again produce a profile plot with the `plot()` method for the resulting object. Argument `which` can be used to specify which of the two grouping variables should be used on the x-axis of the profile plot, with the default being the first grouping variable.

```
R> plot(twoway)
```



The `plot()` method illustrated works similarly to function `line_plot()`. The latter is more generally applicable and can also be used, e.g., for plotting time series.

3.7. χ^2 tests

Function `chisq_test()` implements χ^2 goodness-of-fit tests and χ^2 tests on independence. With the `Eredivisie` data, we can first perform a goodness-of-fit test to see whether the traditional Dutch 4-3-3 system of total football is still reflected in player composition of Dutch football teams. In other words, we test for a multinomial distribution of variable `Position` with the probabilities 1/11, 4/11, 3/11, and 3/11 for goalkeepers, defenders, midfielders, and forwards, respectively.

```
R> chisq_test(Eredivisie, "Position", p = c(1, 4, 3, 3)/11)
```

Position

	Observed N	Expected N	Residual
Goalkeeper	35	37.9	-2.9
Defender	137	151.6	-14.6
Midfielder	121	113.7	7.3
Forward	124	113.7	10.3
Total	417		

Test Statistics

	Position
Chi-Square	3.029[a]
df	3
Asymp. Sig.	.387

a. 0 cells (0.0%) have expected frequencies less than 5. The minimum expected cell frequency is 37.9.

Furthermore, we can test whether the categorical variables `Position` and `Foreign` are independent, i.e., whether the proportions of Dutch and foreign players are the same for all playing positions.

```
R> chisq_test(Eredivisie, c("Position", "Foreign"))
```

Position * Foreign Crosstabulation

			Foreign		Total
			0	1	
Position	Goalkeeper	Count	24	11	35
		Expected Count	23.4	11.6	35.0
	Defender	Count	99	38	137
		Expected Count	91.7	45.3	137.0
	Midfielder	Count	84	37	121
		Expected Count	81.0	40.0	121.0
	Forward	Count	72	52	124
		Expected Count	83.0	41.0	124.0
Total		Count	279	138	417
		Expected Count	279.0	138.0	417.0

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	6.543[a]	3	.088
Likelihood Ratio	6.440	3	.092
N of Valid Cases	417		

a. 0 cells (0.0%) have expected count less than 5. The minimum expected count is 11.6.

3.8. Linear regression

In this section, we compare two regression models to explain the log market values of football players. The first model uses only the player's age as a linear and a squared effect, while the second model adds the remaining contract length and a dummy variable for foreign players.

We first add the squared values of age to the data set.

```
R> Eredivisie$AgeSq <- Eredivisie$Age^2
```

We then estimate the regression models with function `regression()`. As usual in R, we specify the regression models with formulas.

```
R> fit <- regression(logMarketValue ~ Age + AgeSq,
+                    logMarketValue ~ Age + AgeSq + Contract + Foreign,
+                    data = Eredivisie)
R> fit
```

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.260[a]	.068	.063	1.055
2	.453[b]	.206	.198	.976

a. Predictors: (Constant), Age, AgeSq
b. Predictors: (Constant), Age, AgeSq, Contract, Foreign

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	33.193	2	16.596	14.919	.000[b]
	Residual	458.338	412	1.112		
	Total	491.530	414			
2	Regression	101.011	4	25.253	26.513	.000[c]
	Residual	390.519	410	.952		
	Total	491.530	414			

a. Dependent Variable: logMarketValue
b. Predictors: (Constant), Age, AgeSq
c. Predictors: (Constant), Age, AgeSq, Contract, Foreign

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	4.125	1.734		2.379	.018
	Age	.742	.136	2.719	5.456	.000
	AgeSq	-.014	.003	-2.719	-5.457	.000
2	(Constant)	4.007	1.607		2.494	.013
	Age	.684	.126	2.506	5.421	.000
	AgeSq	-.013	.002	-2.417	-5.223	.000
	Contract	.354	.048	.340	7.400	.000
	Foreign	.427	.102	.185	4.185	.000

a. Dependent Variable: logMarketValue

If we only want to print the table containing the model summaries, we can use the argument `statistics` of the `print()` method. In addition, argument `change` can be set to `TRUE` in order to include a test on the change in R^2 from one model to the next.

```
R> print(fit, statistics = "summary", change = TRUE)
```

Model Summary

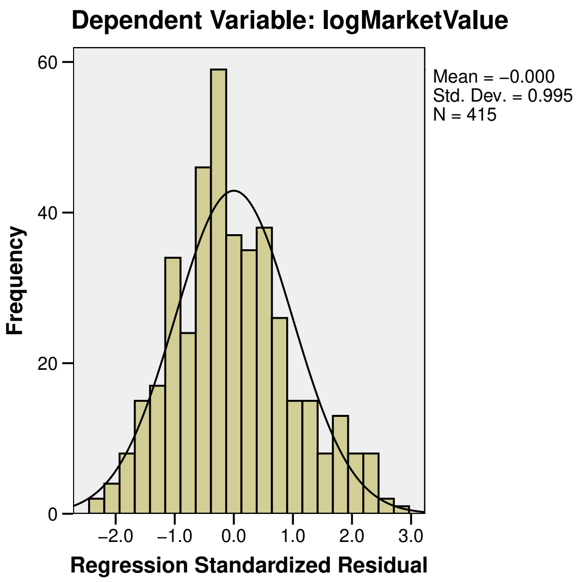
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	.260[a]	.068	.063	1.055	.068	14.919	2	412	.000
2	.453[b]	.206	.198	.976	.138	35.601	2	410	.000

a. Predictors: (Constant), Age, AgeSq
b. Predictors: (Constant), Age, AgeSq, Contract, Foreign

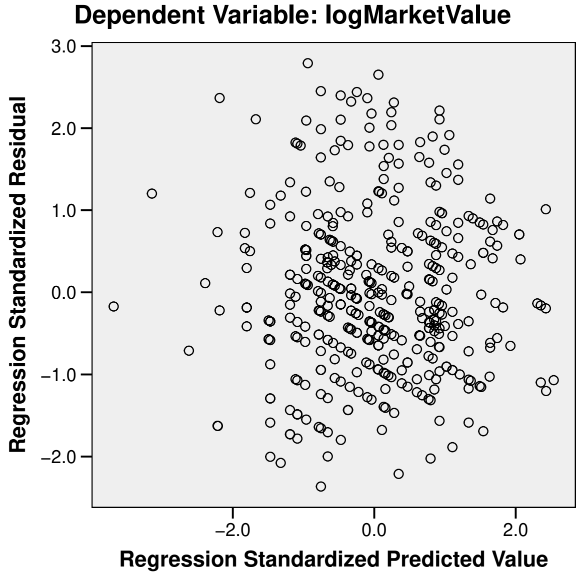
Of course, all `print()` methods for objects returned by functions from package **r2spss** allow to select which tables to print. See the respective help files for details.

The `plot()` method of the regression results can be used to create a histogram of the residuals or a scatter plot of the standardized residuals against the standardized fitted values. Argument `which` can be used to select between those two plots. Mimicking SPSS functionality, the plot is created for the *last* specified model in the call to `regression()`.

```
R> plot(fit, which = "histogram")
```




```
R> plot(fit, which = "scatter")
```



References

Alfons A (2021). ***r2spss**: Format R Output to Look Like SPSS*. R package version 0.2.0, URL https://github.com/aalfons/r2spss/.

Grün B, Zeileis A (2009). "Automatic Generation of Exams in R." *Journal of Statistical Software*, **29**(10), 1–14. doi:10.18637/jss.v029.i10.

IBM Corp (2021). *IBM SPSS Statistics, Version 28.0.*

Mittelbach F, Goossens M, Braams J, Carlisle D, Rowley C (2004). *The LaTeX Companion*. 2nd edition. Addison-Wesley, Boston, MA. ISBN 0-201-36299-6.

Pantigny F (2021). *The Package nicematrix*. LaTeX package version 6.4, URL https://CTAN.org/pkg/nicematrix.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Xie Y (2015). *Dynamic Documents with R and **knitr***. 2nd edition. Chapman and Hall/CRC, Boca Raton, FL. ISBN 978-1498716963, URL https://yihui.org/knitr/.

Xie Y (2021). ***knitr**: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.36, URL https://CRAN.R-project.org/package=knitr.

Zeileis A, Grün B, Leisch F, Umlauf N (2020). ***exams**: Automatic Generation of Exams in R*. R package version 2.3-6, URL https://CRAN.R-project.org/package=exams.

Zeileis A, Umlauf N, Leisch F (2014). "Flexible Generation of E-Learning Exams in R: **Moodle** Quizzes, **OLAT** Assessments, and Beyond." *Journal of Statistical Software*, **58**(1), 1–36. doi:10.18637/jss.v058.i01.

Affiliation:

Andreas Alfons
Econometric Institute
Erasmus School of Economics
Erasmus University Rotterdam
PO Box 1738
3000DR Rotterdam, The Netherlands
E-mail: alfons@ese.eur.nl
URL: https://personal.eur.nl/alfons/