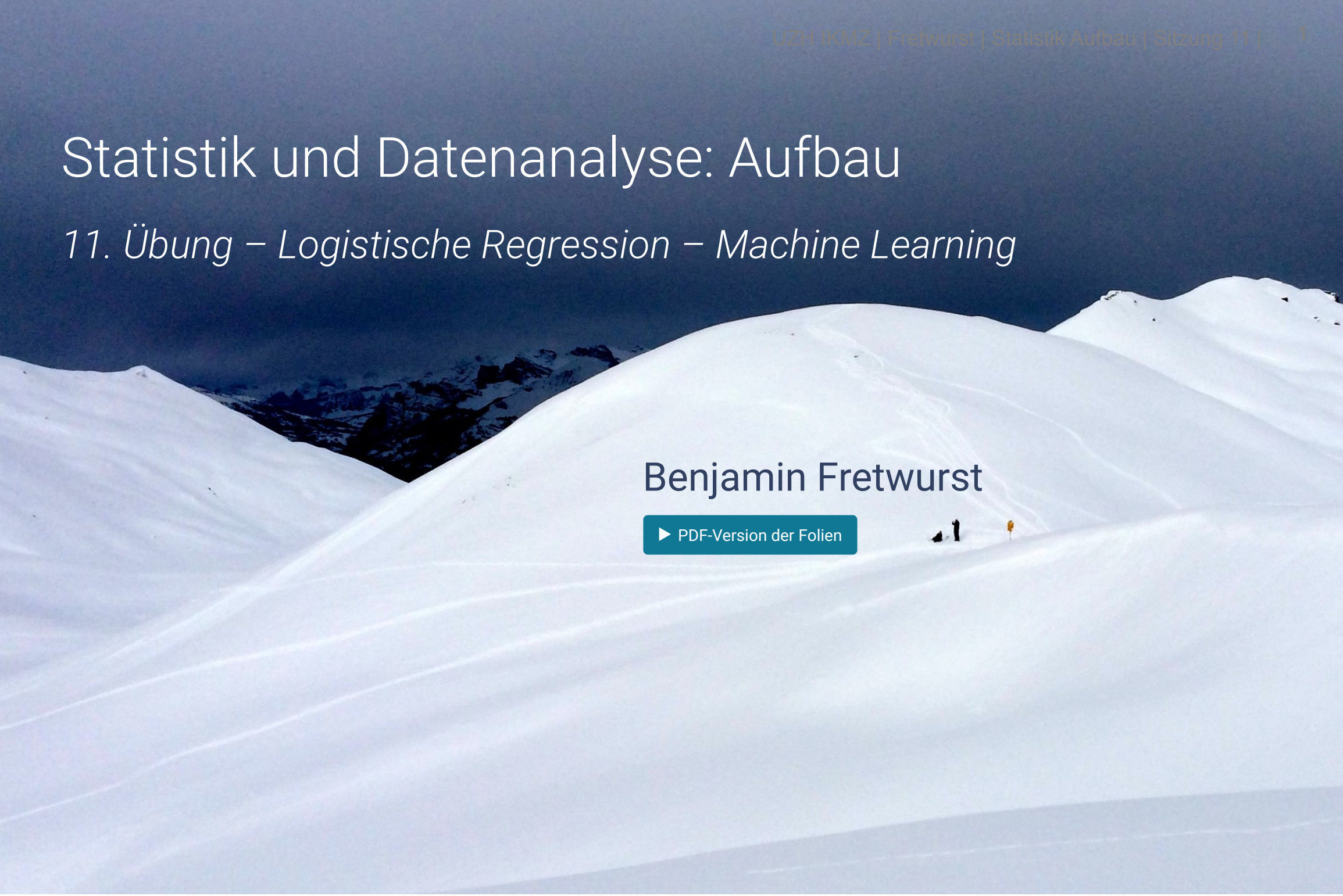
Statistik und Datenanalyse: Aufbau

11. Übung – Logistische Regression – Machine Learning



Benjamin Fretwurst

▶ PDF-Version der Folien

Inhalt

Orga

Letzte Sitzung am 20.12.2023 ist online only. Es gibt dann einen Zusammenfassung-/Klausurvorbereitungsonlinevodcast.

Photo courtesy of Lisa Vogel

Lernziele

Übung in logistischer Regression und ML

- Übung 4 und Text Zerback Wirz! lesen! (liegt beides auch in OLAT unter Materialien bzw. Texte)

1 Logistische Regression

RAQ erklärt Interesse an Maschine Learning

LogReg: Aversion gegen Computer erklärt Interesse an ML

Characteristic	OR[1]	95% CI[1]	p-value	VIF[1]
ML1	0.74	0.53, 1.02	0.065	1.0
ML2	0.54	0.34, 0.83	0.007	1.7
ML3	1.74	1.12, 2.81	0.018	1.7

[1] OR = Odds Ratio, CI = Confidence Interval, VIF = Variance Inflation Factor

Null deviance = 230; Null df = 165; Log-likelihood = -109; AIC = 226; BIC = 238; Deviance = 218; Residual df = 162; No. Obs. = 166

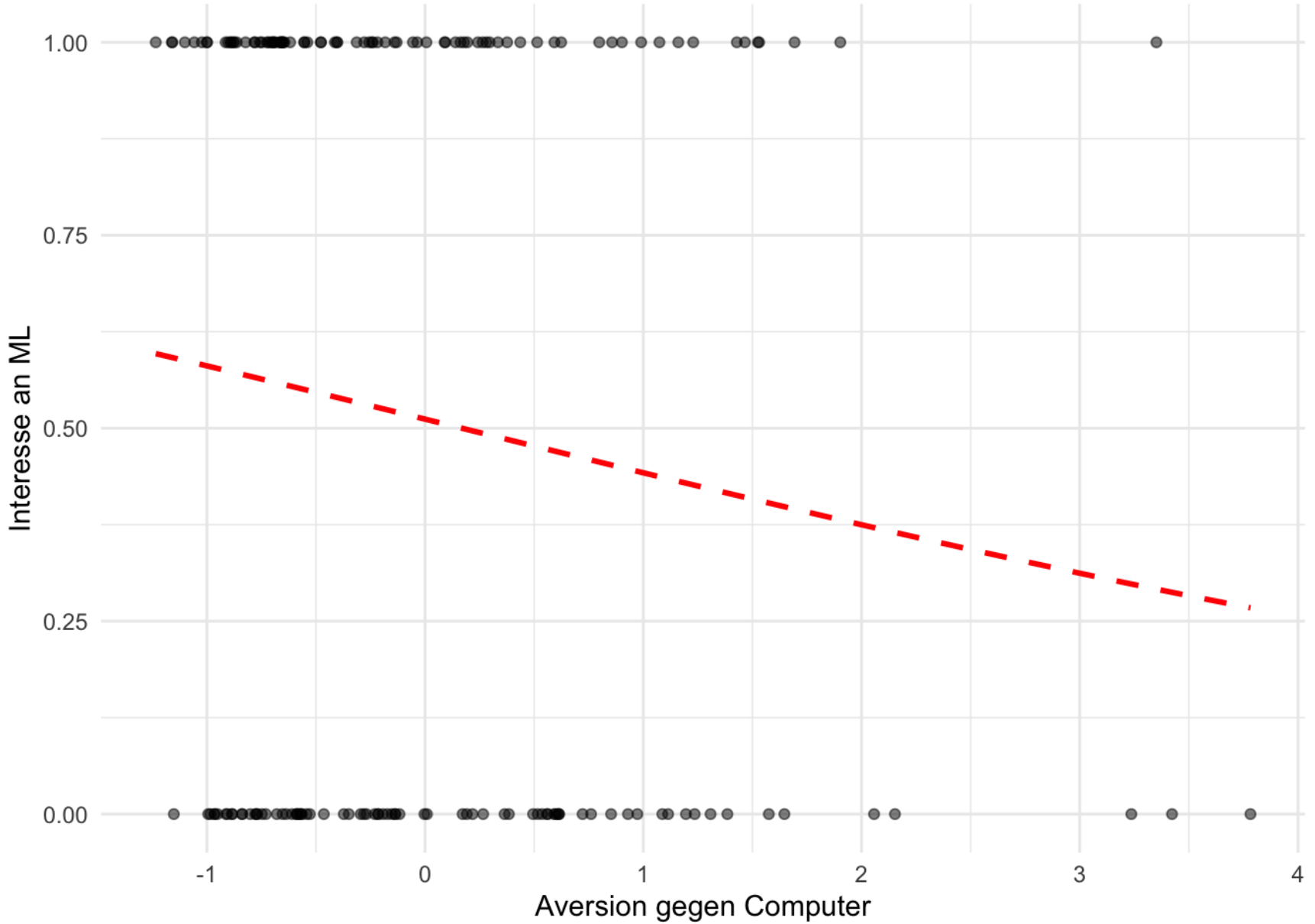
LM: Aversion gegen Computer erklärt Interesse an ML

Characteristic	Beta	95% CI[1]	p-value	VIF[1]
ML1	-0.07	-0.15, 0.00	0.066	1.0
ML2	-0.14	-0.24, -0.04	0.005	1.6
ML3	0.13	0.02, 0.23	0.016	1.6

[1] CI = Confidence Interval, VIF = Variance Inflation Factor

R^2 = 0.071; Adjusted R^2 = 0.054; Sigma = 0.488; Statistic = 4.13; p-value = 0.007; df = 3; Log-likelihood = -114; AIC = 239; BIC = 254; Deviance = 38.5; Residual df = 162; No. Obs. = 166

RAQ-ML-«S-Kurve» sieht gerade aus



1.1 Multinominale Regression

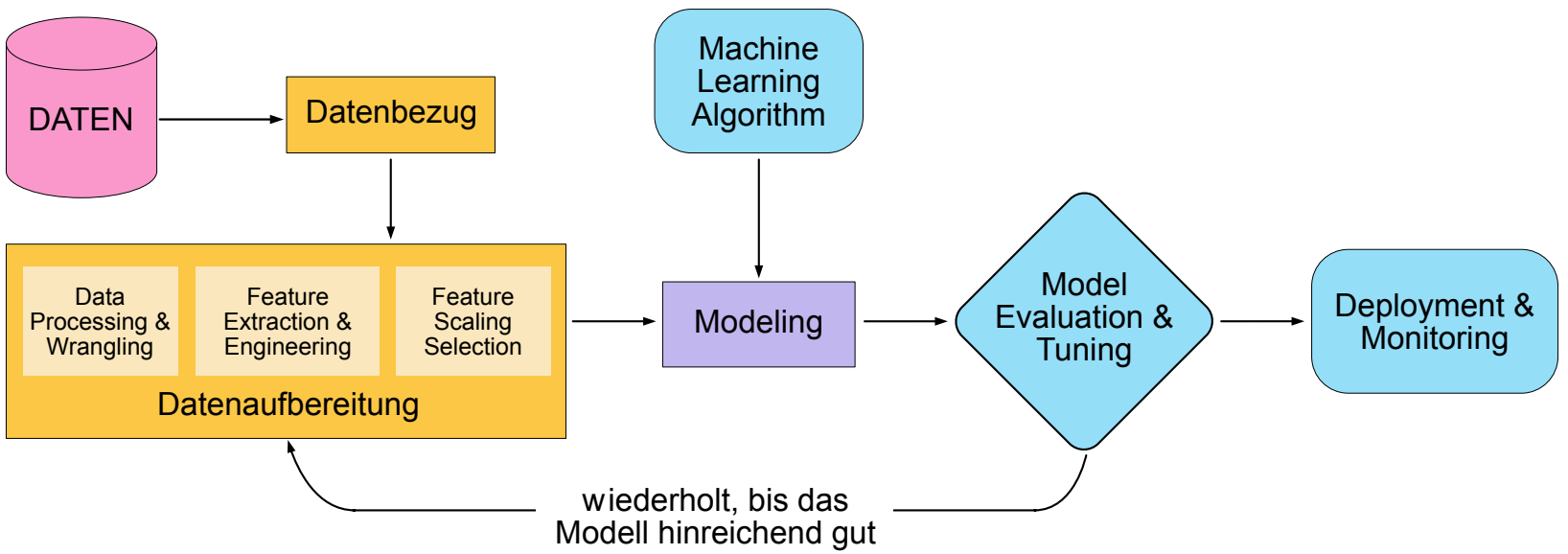
Multinomial bedeutet, dass es eine kategoriale (multinomiale) AV gibt. Im Prinzip werden mehrere binäre logistische Regressionen durchgeführt und zusätzlich ausgegeben, wie gut das Gesamtmodel ist. Es kommen die Fit-Kennungen AIC und BIC dazu.

Supervised ML

Trainieren ➪ Testen ➪ Anwenden

Die verfügbaren Daten werden in einen grösseren und einen kleineren Teil getrennt:

1. Trainingsdaten, an denen das Modell trainiert wird (Modelling)
2. Testdaten, an denen das Modell evaluiert wird.



Übung 4

In der Übung beschäftigen wir uns mit Daten des Untergangs der Titanic. Schauen Sie sich jeweils die Befehle an und finden Sie heraus, was die Befehle machen. Machen Sie sich Notizen, wenn Sie etwas nicht verstehen.

Daten einlesen

Legen Sie für diese Übung einen Ordner an, erstellen Sie dort eine qmd und kopieren Sie die Folgenden Daten in einen dortigen Unterordner «data»:

Suchen und nehmen Sie den Datensatz “train.csv”: Downlaod der Daten: https://www.kaggle.com/competitions/titanic/data

▶ Code

1.2 Datenaufbereitung

▶ Code

… und mal die Daten angucken:

▶ Code

PassengerId Survived Pclass Name
Min. : 1.0 Min. :0.0000 Min. :1.000 Length:891
1st Qu.:223.5 1st Qu.:0.0000 1st Qu.:2.000 Class :character
Median :446.0 Median :0.0000 Median :3.000 Mode :character
Mean :446.0 Mean :0.3838 Mean :2.309
3rd Qu.:668.5 3rd Qu.:1.0000 3rd Qu.:3.000
Max. :891.0 Max. :1.0000 Max. :3.000

```
Sex                 Age             SibSp           Parch
```

Length:891 Min. : 0.42 Min. :0.000 Min. :0.0000
Class :character 1st Qu.:20.12 1st Qu.:0.000 1st Qu.:0.0000
Mode :character Median :28.00 Median :0.000 Median :0.0000
Mean :29.70 Mean :0.523 Mean :0.3816
3rd Qu.:38.00 3rd Qu.:1.000 3rd Qu.:0.0000
Max. :80.00 Max. :8.000 Max. :6.0000
NA's :177
Ticket Fare Cabin Embarked
Length:891 Min. : 0.00 Length:891 S :644
Class :character 1st Qu.: 7.91 Class :character C :168
Mode :character Median : 14.45 Mode :character Q : 77
Mean : 32.20 NA's: 2
3rd Qu.: 31.00
Max. :512.33

Kinder Age_z Pclass_f Cabin_D
Mode :logical Min. :-29.279 3:491 Min. :0.000

FALSE:643 1st Qu.: -9.574 2:184 1st Qu.:0.000
TRUE :71 Median : -1.699 1:216 Median :0.000
NA’s :177 Mean : 0.000 Mean :0.229
3rd Qu.: 8.301 3rd Qu.:0.000
Max. : 50.301 Max. :1.000
NA’s :177

Die PassengerId ist einefach eine Identifikationsnummer.

- Es gibt dann eine Variable, die “Survived” heisst, die ein Minimum von 0 hat und ein Maximum von 1. Das deutet sehr auf eine Dummy hin. Da der Durchschnitt (“Mean”) = 0.38 ist, wissen wir jetzt schon, dass 38 Prozent der Passagiere überlebt haben (der Mittelwert einer Dummy ist immer der Prozentsatz der 1er-Gruppe).
- Dann kommt noch der Name als Zeichenvariable,
- das Alter, das von 0.42 bis 80 geht. Von 177 Personen fehlen die Altersangaben.

Informieren Sie sich über die übrigen Variablen auf (https://www.kaggle.com/competitions/titanic/data)[«Kaggle»].

Daten in Trainings- und Testdaten aufteilen

Was passiert hier?

▶ Code

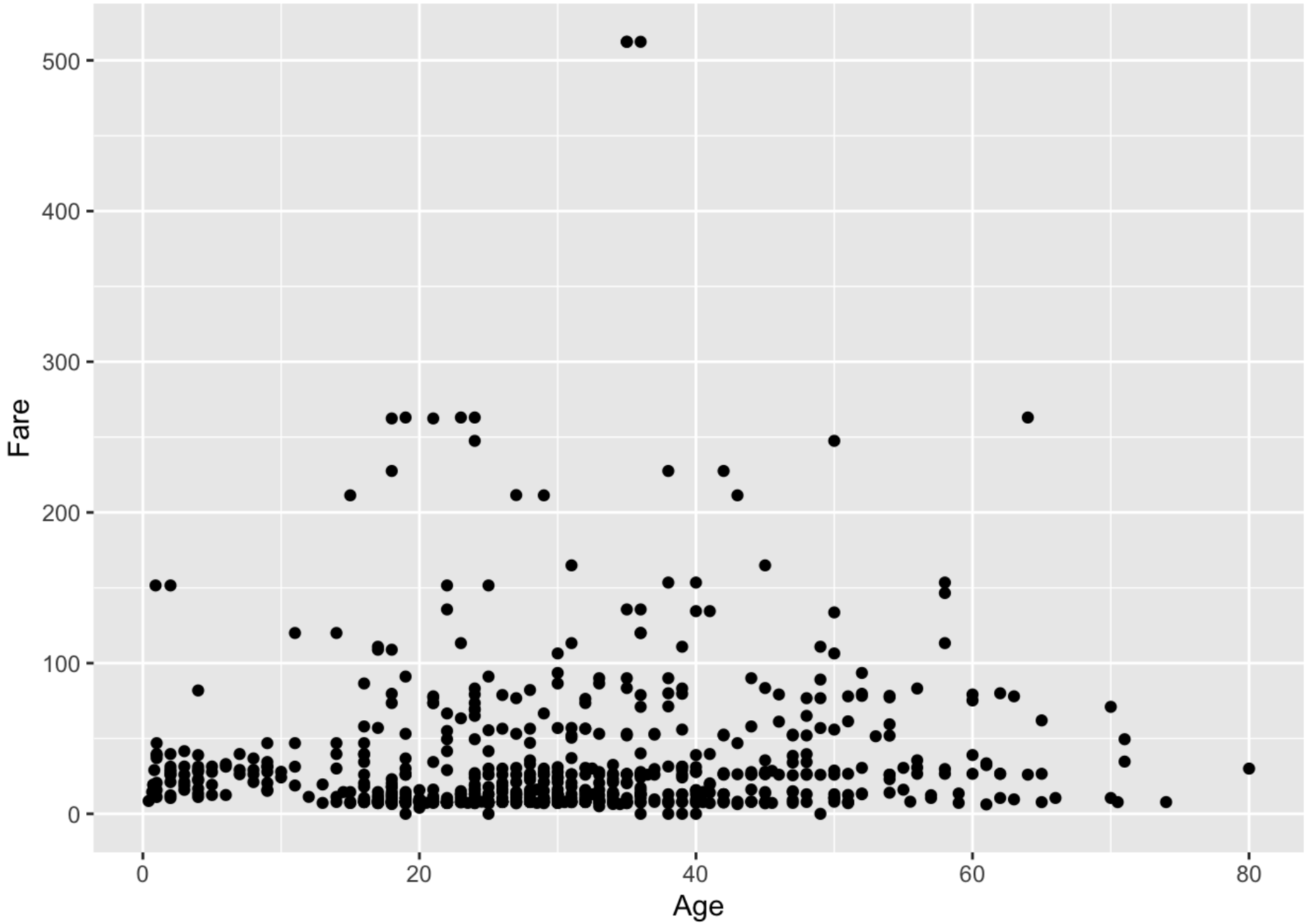
Sehen kann man nach diesem r-Chunk übrigens nichts, weil nur Datensätze im Hintergrund aufgeteilt wurden. Also suchen wir mal nach guten Datenvisualisierungen.

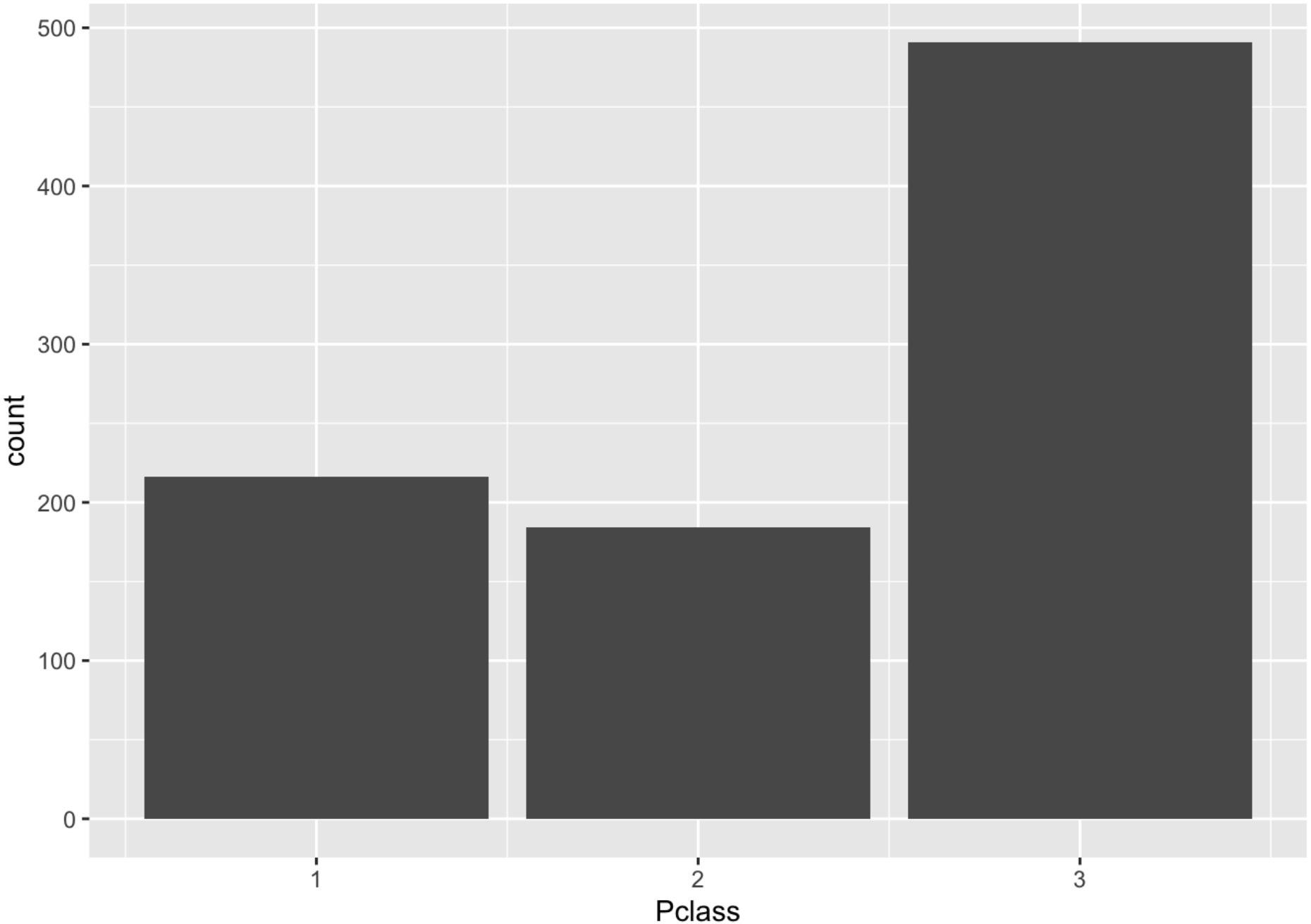
Übung: Experimentieren Sie mit verschiedenen Zahlen in set.seed() und grösseren und kleineren Werten in sample_frac().

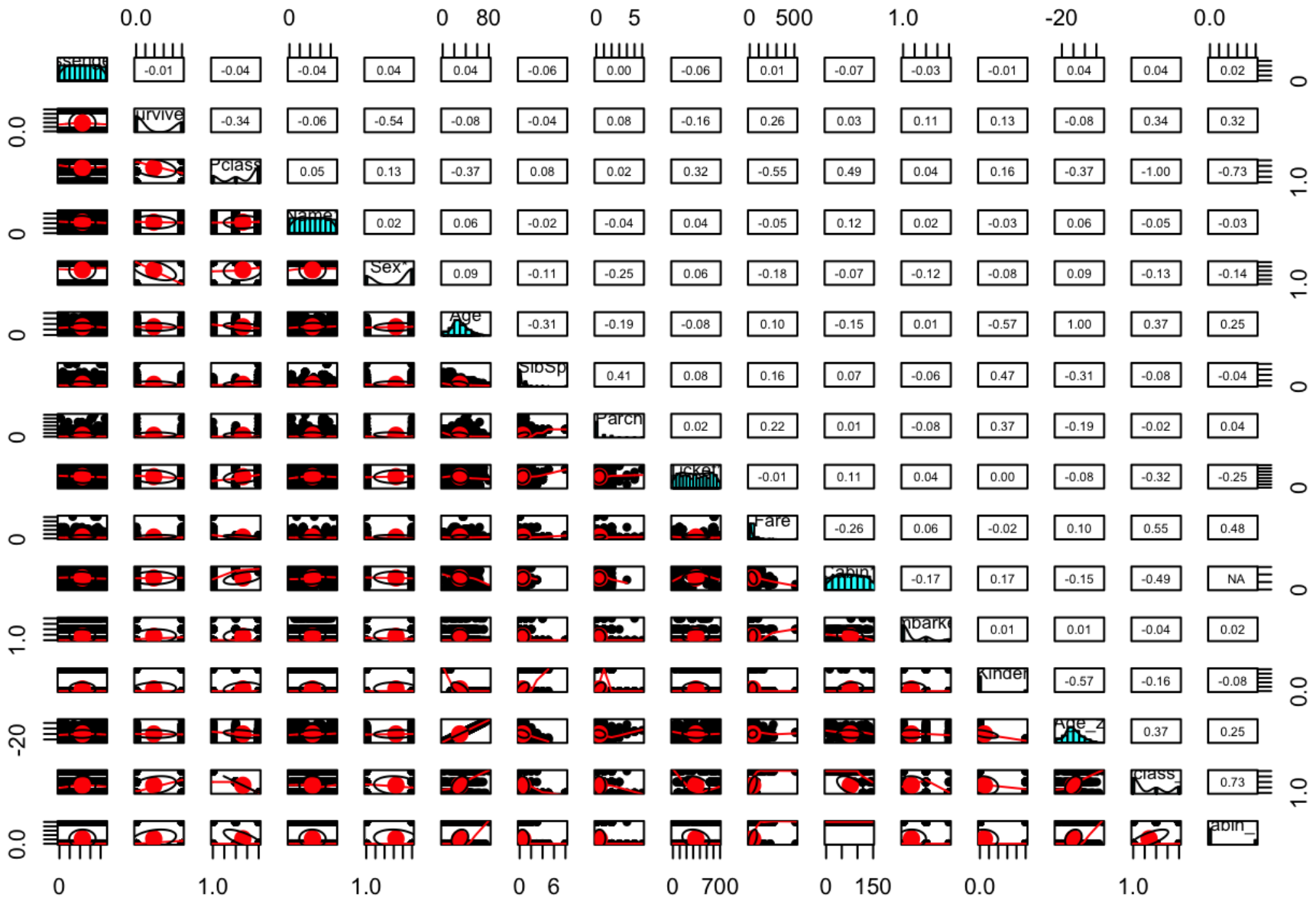
Datenvisualisierung

Eine einfache Darstellungsmöglichkeit ist ein sogenannter “Scatterplot” der die Lage von Fällen in ein Koordinatensystem einteilt, das durch zwei Variablen gebildet wird. Im Beispiel (Abbildung @ref(titanic-Datenvisualisierungen)) ist es das Alter der Passagiere “Age” und der Fahrpreis “Fare”. Als Zweites haben wir ein Balkendiagramm für die Passagierklassen. Im letzten sehr aufwendigen Plotgrafik werden die Zweierbeziehungen aller Variablen dargestellt, also wie sie miteinander korrelieren (obere Nebendiagonalen), wie ihre Verteilung ist (auf der Diagonale mit Namen) und wie ihre gemeinsame Streuung ist, also ein Scatterplot in der unteren Nebendiagonalen. Mehr zu diesen SPLOM finden Sie hier: https://cran.r-project.org/web/packages/psych/vignettes/intro.pdf.

▶ R-Code


500
400
300
200
100
0
Fare
0
20
40
60
80
Age


500
400
300
200
100
0
count
1
2
3
Pclass


0.0
0
0 80
0 5
0 500
1.0
-20
0.0
-0.01 -0.04 -0.04 0.04 0.04 -0.06 0.00 -0.06 0.01 -0.07 -0.03 -0.01 0.04 0.04 0.02
urvive -0.34 -0.06 -0.54 -0.08 -0.04 0.08 -0.16 0.26 0.03 0.11 0.13 -0.08 0.34 0.32
Pclass 0.05 0.13 -0.37 0.08 0.02 0.32 -0.55 0.49 0.04 0.16 -0.37 -1.00 -0.73
Name 0.02 0.06 -0.02 -0.04 0.04 -0.05 0.12 0.02 -0.03 0.06 -0.05 -0.03
Sex 0.09 -0.11 -0.25 0.06 -0.18 -0.07 -0.12 -0.08 0.09 -0.13 -0.14
Age -0.31 -0.19 -0.08 0.10 -0.15 0.01 -0.57 1.00 0.37 0.25
SibSp 0.41 0.08 0.16 0.07 -0.06 0.47 -0.31 -0.08 -0.04
Parch 0.02 0.22 0.01 -0.08 0.37 -0.19 -0.02 0.04
Ticket -0.01 0.11 0.04 0.00 -0.08 -0.32 -0.25
Fare -0.26 0.06 -0.02 0.10 0.55 0.48
Cabin -0.17 0.17 -0.15 -0.49 NA
Embark 0.01 0.01 -0.04 0.02
Kinder -0.57 -0.16 -0.08
Age_z 0.37 0.25
Class 0.73
Cabin_
0 1.0 1.0 0 6 0 700 0 150 0.0 1.0

Modellbildung für den Fahrpreis

Übung: Experimentieren Sie mit dem Syntax! Kopieren Sie sich die Zeile für das Modell, löschen von “Age_z” bis “Kinder” alles heraus und schätzen Sie mal. Schauen Sie sich das Ergebnis gut an und achten Sie darauf, was passiert, wenn Sie die Summanden für I(Age_z^2) usw. wieder in das Modell tun. Am Ende können Sie versuchen das Modell durch weitere Variablen ergänzen und verbessern oder andere Teile wieder herausnehmen. Manche Variablen wurden erst noch erstellt (zB “Kinder” oder “Age_z”). Die entsprechende Datenaufbereitung.qmd können Sie hier abrufen .

Übung: Interpretieren Sie den Output!

▶ Code

term	estimate	std.error	statistic	p.value
(Intercept)	2.657	0.076	35.047	0.000
Sexmale	-0.316	0.070	-4.520	0.000
Age_z	-0.003	0.003	-1.009	0.314
I(Age_z^2)	0.000	0.000	-0.406	0.685
Survived	-0.077	0.073	-1.050	0.294
Pclass_f2	0.492	0.072	6.859	0.000
Pclass_f1	1.778	0.077	23.104	0.000
KinderTRUE	0.607	0.173	3.515	0.000

▶ Code

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.58	0.57	0.64	103.29	0	7	-517.85	1053.7	1092.26	216.7	528	536

Regressionsoutput mit `sjPlot::tab_model`

▶ Code

	log(Fare+1)					
Predictors	*b*	*std. b*	*CI*	*standardized CI*	*p*	std. p
(Intercept)	2.66	0.83	2.51 – 2.81	0.80 – 0.87	**<0.001**	**<0.001**
Sex [male]	-0.32	-0.06	-0.45 – -0.18	-0.10 – -0.02	**<0.001**	**0.003**
Age z	-0.00	-0.02	-0.01 – 0.00	-0.05 – 0.00	0.314	0.063
Age z^2	-0.00	-0.00	-0.00 – 0.00	-0.02 – 0.01	0.685	0.639
Survived	-0.08	-0.00	-0.22 – 0.07	-0.02 – 0.02	0.294	0.914
Pclass f [2]	0.49	0.06	0.35 – 0.63	0.02 – 0.10	**<0.001**	**0.004**
Pclass f [1]	1.78	0.44	1.63 – 1.93	0.39 – 0.48	**<0.001**	**<0.001**
KinderTRUE	0.61	0.05	0.27 – 0.95	-0.04 – 0.15	**<0.001**	0.272
Observations	536					
R^2 / R^2 adjusted	0.578 / 0.572					

Regressionsoutput mit `gtsummary::tbl_regression`

Übung: Was für ein Problem sehen Sie im folgenden Output? Bei welcher Befehlszeile müssten Sie das # wegnehmen, um das Problem zu lösen?

▶ Code

Characteristic	Beta	95% CI[1]	p-value	GVIF[1]	Adjusted GVIF[2,1]
Sex				1.5	1.2
female	—	—			
male	-0.32	-0.45, -0.18	<0.001		
Age_z	0.00	-0.01, 0.00	0.3	3.1	1.8
I(Age_z^2)	0.00	0.00, 0.00	0.7	2.5	1.6
Survived	-0.08	-0.22, 0.07	0.3	1.7	1.3
Pclass_f				1.4	1.1
3	—	—			
2	0.49	0.35, 0.63	<0.001		
1	1.8	1.6, 1.9	<0.001		

R^2 = 0.578; Adjusted R^2 = 0.572; Sigma = 0.641; Statistic = 103; p-value = <0.001; df = 7; Log-likelihood = -518; AIC = 1,054; BIC = 1,092; Deviance = 217; Residual df = 528; No. Obs. = 536

Characteristic	Beta	95% CI[1]	p-value	GVIF[1]	Adjusted GVIF[2,1]
Kinder				3.5	1.9
FALSE	—	—			
TRUE	0.61	0.27, 0.95	<0.001		

[1] CI = Confidence Interval, GVIF = Generalized Variance Inflation Factor

[2] GVIF^[1/(2*df)]

R^2 = 0.578; Adjusted R^2 = 0.572; Sigma = 0.641; Statistic = 103; p-value = <0.001; df = 7; Log-likelihood = -518; AIC = 1,054; BIC = 1,092; Deviance = 217; Residual df = 528; No. Obs. = 536

Voraussetzungschecks

Übung: Suchen Sie die Befehle und führen Sie sie aus:

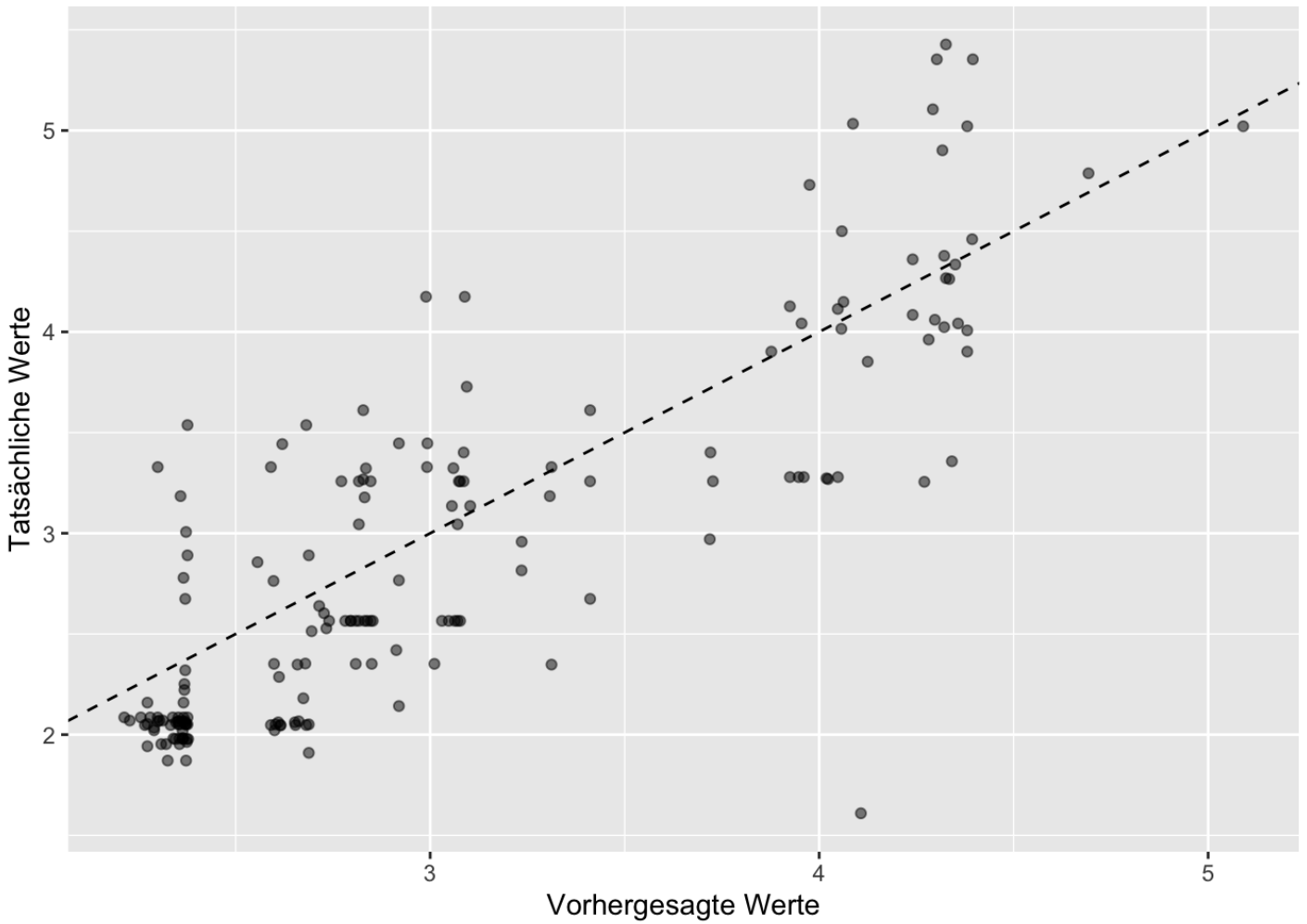
Die möglichen Verletzungen der Voraussetzungen sind:

- Multikollinearität
- Deutliche Abweichung von der Normalverteilung in den Fehlern
- Heteroskedastizität
- Outlier

Algorithmus für den Fahrpreis

Prognosewerte erstellen:

- Code



Überlebensprognose

Zweite Analyse: Überlebensprognose (Dummy) mit linearer Regression. Was denken Sie?

▶ Code

term	estimate	std.error	statistic	p.value
(Intercept)	0.579	0.038	15.427	<0.001
Pclass_f2	0.180	0.042	4.285	<0.001
Pclass_f1	0.376	0.043	8.772	<0.001
Sexmale	-0.508	0.035	-14.392	<0.001
Age_z	-0.003	0.002	-1.464	0.144
I(Age_z^2)	0.000	0.000	-0.184	0.854
KinderTRUE	0.125	0.103	1.217	0.224

▶ Code

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.403	0.397	0.382	59.587	<0.001	6	-241.157	498.314	532.587	77.177	529	536

Logistische Regression

▶ Code

term	estimate	std.error	statistic	p.value
(Intercept)	0.235	0.221	1.065	0.287
Pclass_f2	1.130	0.286	3.946	<0.001
Pclass_f1	2.158	0.287	7.518	<0.001
Sexmale	-2.703	0.244	-11.087	<0.001
KinderTRUE	1.173	0.373	3.142	0.002

▶ Code

null.deviance	df.null	logLik	AIC	BIC	deviance	df.residual	nobs
724.287	535	-241.935	493.871	515.291	483.871	531	536

Kennwerte

Übung: Interpretieren Sie den Output

▶ Code

Überlebensanalyse zum Titanicunglück mit sjPlot

	Survived		
Predictors	*Odds Ratios*	*CI*	*p*
(Intercept)	1.27	0.82 – 1.96	0.287
Pclass f [2]	3.10	1.77 – 5.47	**<0.001**
Pclass f [1]	8.65	4.99 – 15.41	**<0.001**
Sex [male]	0.07	0.04 – 0.11	**<0.001**
KinderTRUE	3.23	1.56 – 6.78	**0.002**
Observations	536		
R^2 Tjur	0.404		

Wieder auch mit `gtsummary::`

▶ Code

Überlebensanalyse zum Titanicunglück mit gtsummary

Characteristic	OR[1]	95% CI[1]	p-value	GVIF[1]	Adjusted GVIF[2,1]
Pclass_f				1.2	1.0
3	—	—			
2	3.10	1.77, 5.47	<0.001		
1	8.65	4.99, 15.4	<0.001		
Sex				1.1	1.0
female	—	—			
male	0.07	0.04, 0.11	<0.001		
Kinder				1.1	1.0
FALSE	—	—			
TRUE	3.23	1.56, 6.78	0.002		

[1] OR = Odds Ratio, CI = Confidence Interval, GVIF = Generalized Variance Inflation Factor

[2] GVIF^[1/(2*df)]

Null deviance = 724; Null df = 535; Log-likelihood = -242; AIC = 494; BIC = 515; Deviance = 484; Residual df = 531; No. Obs. = 536

Voraussetzungschecks

Übung: Führen Sie die Checks für die Voraussetzungen aus!

Vorhersagetest

Was sagt dieser Test?

- Code

Take Home – Ausblick – Vokabeln

Take Home

Logistische Regression

- Sie können eine logistische Regression in R rechnen und interpretieren
- Sie können Machine Learning in R anwenden

Ausblick

Sie lernen Clusteranalysen kennen. Wie man also Gruppen anhand von Eigenschaften identifiziert