

Rによるデータ解析

第1講 - さまざまな多変量解析とR言語の使い方

村田 昇

R言語の概要

R言語とは

- 統計計算のための言語と環境の総称
- オープンソース・フリーソフトウェア
- **パッケージ** を利用して容易に機能拡張が可能
 - パッケージの開発は非常に活発 (現在 10000 を越える)
 - 最新の技術や方法が簡単に導入できることも多い
- プロジェクトのサイト
<https://www.r-project.org/>

RStudio とは

- Posit 社が開発している統合開発環境 (IDE)
 - R によるデータ解析や統計計算・パッケージ開発を支援
 - OS に依存しない対話型操作環境を提供
- 本講義では RStudio を用いて説明を行う
- Posit Software のサイト
<https://posit.co>

R言語の得意分野

- **データの分類・集計・整理**
 - 記述統計量 (基本・要約統計量) の計算
 - グラフによる視覚化
 - さまざまな統計分析 (多変量解析を含む)
- プログラムによる **処理の自動化**
- **確率的シミュレーション** (モンテカルロ法)
 - 擬似乱数による不確定性を含む現象の模擬

参考：R言語のオンラインコース

- RStudio の初心者向ガイド
<https://education.rstudio.com/learn/beginner/>
- posit Cloud の自習コース
<https://posit.cloud/learn/recipes/>
- Data Camp の R 入門コース
<https://onboarding.datacamp.com/>

多変量解析

多変量解析とは

- 複数の変量からなるデータを分析する手法の総称
 - 回帰分析：複数の量を用いて注目する変数の値を説明する
 - 主成分分析：全体を説明する少数の特徴量を構成する
 - 判別分析：特徴量の違いでカテゴリ分けを行う
 - クラスタ分析：特徴量の違いに着目してクラスを構成する
 - 時系列解析：時間とともに変化する現象を記述する
- 機械学習で使われる手法の基礎
 - 教師あり問題：回帰分析 (量的データ)・判別分析 (質的データ)
 - 教師なし問題：主成分分析・クラスタ分析

回帰分析の考え方

- ある変数 (目的変数) を別の変数 (説明変数) によって説明・予測するための関係式 (回帰式) を構成する
 - 単回帰：一つの変数で目的変数を説明する
 - 重回帰：複数の変数で目的変数を説明する
- 分析の事例
 - 広告宣伝費と商品の売上を予測する式を作り，広告効果があるかどうか判定する
 - 築年数・駅からの距離・広さ・間取りで家賃を説明する式を作り，新規に家賃を設定する際に利用する

例：体重と脳の重さの関係

	body	brain
Mountain beaver	1.350	8.1
Cow	465.000	423.0
Grey wolf	36.330	119.5
Goat	27.660	115.0
Guinea pig	1.040	5.5
Dipliodocus	11700.000	50.0
Asian elephant	2547.000	4603.0
Donkey	187.100	419.0
Horse	521.000	655.0
Potar monkey	10.000	115.0
Cat	3.300	25.6
Giraffe	529.000	680.0
Gorilla	207.000	406.0
Human	62.000	1320.0
African elephant	6654.000	5712.0
Triceratops	9400.000	70.0
Rhesus monkey	6.800	179.0
Kangaroo	35.000	56.0
Golden hamster	0.120	1.0
Mouse	0.023	0.4
Rabbit	2.500	12.1
Sheep	55.500	175.0
Jaguar	100.000	157.0
Chimpanzee	52.160	440.0
Rat	0.280	1.9
Brachiosaurus	87000.000	154.5
Mole	0.122	3.0
Pig	192.000	180.0

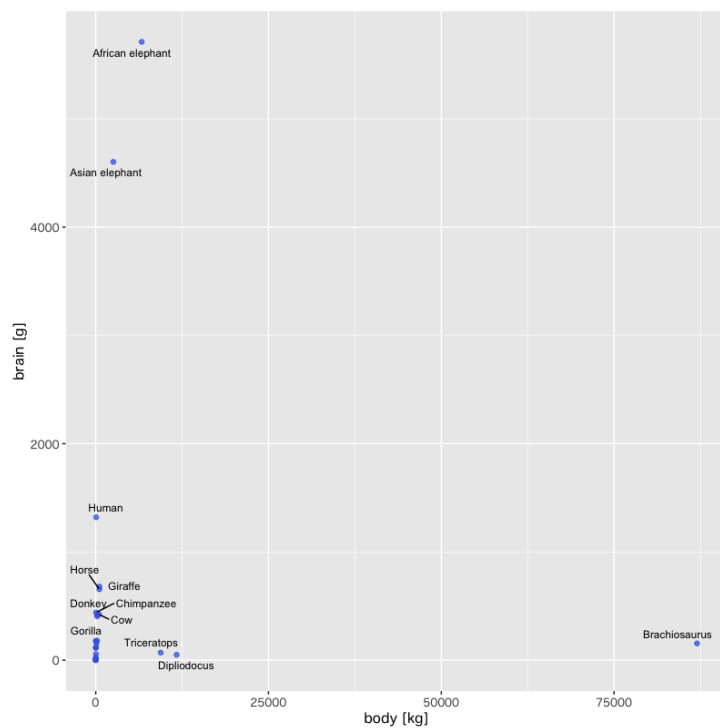


Figure 1: 体重と脳の重さの関係

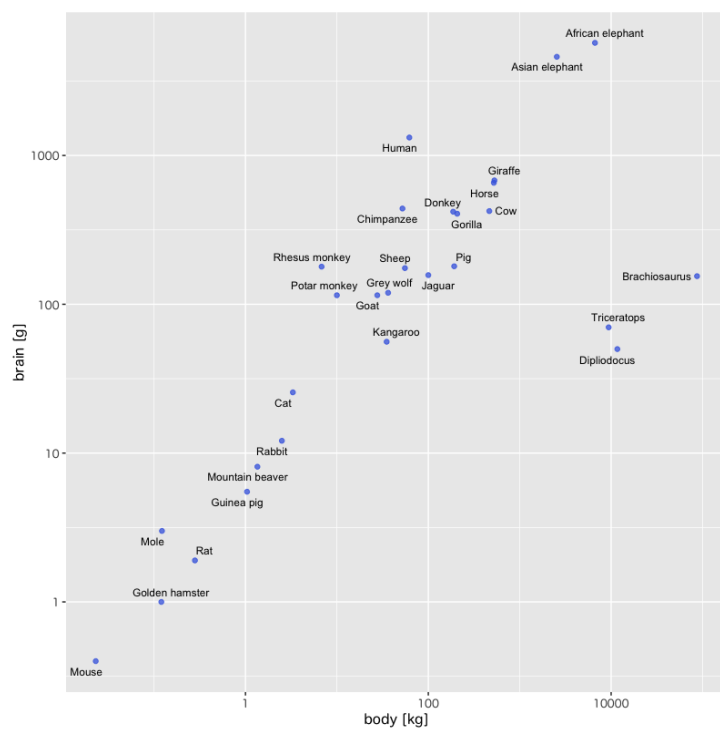


Figure 2: 体重と脳の重さの関係 (対数変換)

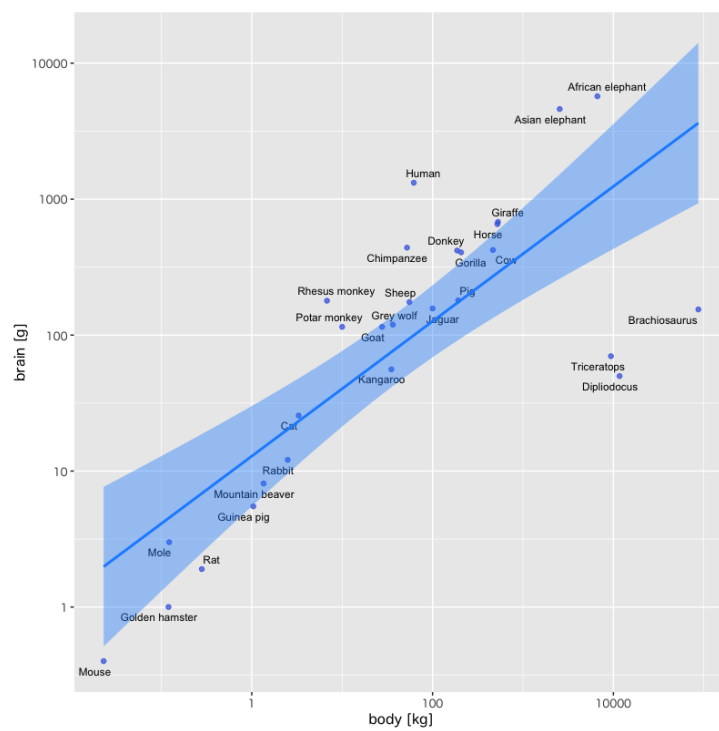


Figure 3: 回帰式とその信頼区間

例：ボルドーワインの価格と気候の関係

VINT	LPRICE2	WRAIN	DEGREES	HRAIN	TIME_SV
1952	-0.99868	600	17.1167	160	31
1953	-0.45440	690	16.7333	80	30
1954	NA	430	15.3833	180	29
1955	-0.80796	502	17.1500	130	28
1956	NA	440	15.6500	140	27
1957	-1.50926	420	16.1333	110	26
1958	-1.71655	582	16.4167	187	25
1959	-0.41800	485	17.4833	187	24
1960	-1.97491	763	16.4167	290	23
1961	0.00000	830	17.3333	38	22
1962	-1.10572	697	16.3000	52	21
1963	-1.78098	608	15.7167	155	20
1964	-1.18435	402	17.2667	96	19
1965	-2.24194	602	15.3667	267	18
1966	-0.74943	819	16.5333	86	17
1967	-1.65388	714	16.2333	118	16
1968	-2.25018	610	16.2000	292	15
1969	-2.14784	575	16.5500	244	14
1970	-0.90544	622	16.6667	89	13
1971	-1.30031	551	16.7667	112	12
1972	-2.28879	536	14.9833	158	11
1973	-1.85700	376	17.0667	123	10
1974	-2.19958	574	16.3000	184	9
1975	-1.20168	572	16.9500	171	8
1976	-1.37264	418	17.6500	247	7
1977	-2.23503	821	15.5833	87	6
1978	-1.30769	763	15.8167	51	5
1979	-1.53960	717	16.1667	122	4
1980	-1.99582	578	16.0000	74	3
1981	NA	535	16.9667	111	2
1982	NA	712	17.4000	162	1
1983	NA	845	17.3833	119	0
1984	NA	591	16.5000	119	-1
1985	NA	744	16.8000	38	-2
1986	NA	563	16.2833	171	-3
1987	NA	452	16.9833	115	-4
1988	NA	808	17.1000	59	-5
1989	NA	443	NA	82	-6

Characteristic	Beta	95% CI	p-value
WRAIN	0.0012	0.0002, 0.0022	0.024
DEGREES	0.6164	0.4190, 0.8138	<0.001
HRAIN	-0.0039	-0.0055, -0.0022	<0.001
TIME_SV	0.0238	0.0090, 0.0387	0.003

Abbreviation: CI = Confidence Interval
 $R^2 = 0.828$; Adjusted $R^2 = 0.796$; Statistic = 26.4; p-value = <0.001

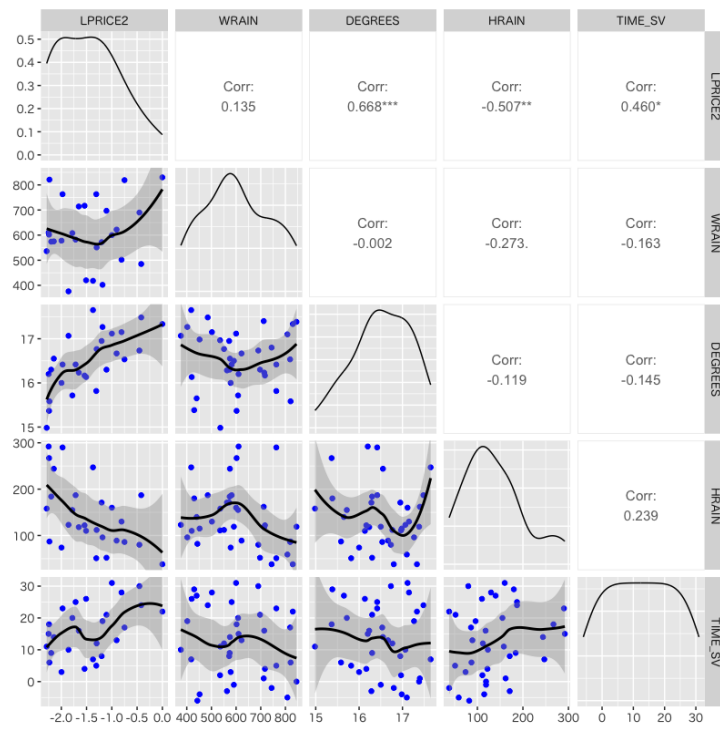


Figure 4: 価格と気候の散布図

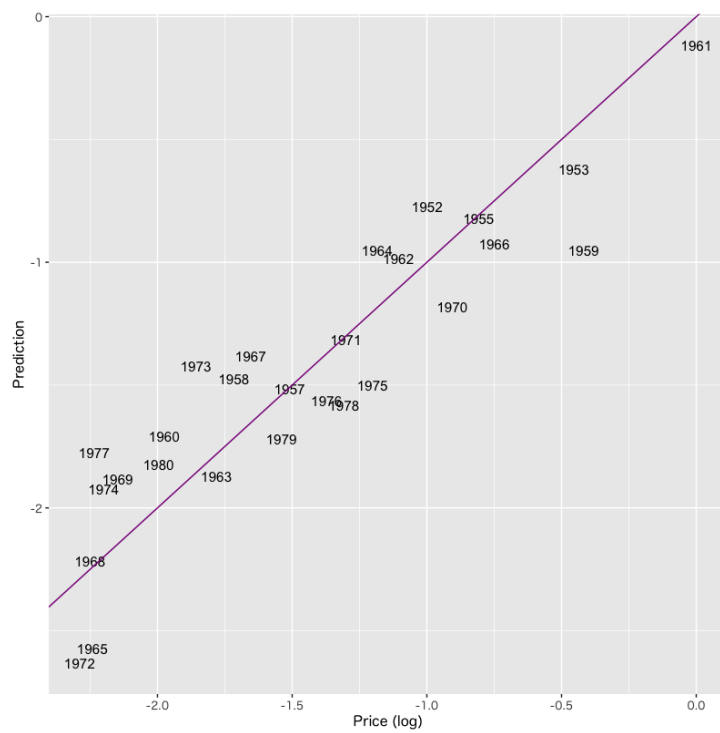


Figure 5: 重回帰による予測値と実際の価格

主成分分析の考え方

- 多数の変数を与えられたときに、変数のもつ構造を効率的に記述できる少数個の特徴量を構成する
- 分析の事例
 - 野球選手の打撃成績 (打率, 本塁打数, 打点など) から、打者としての特徴を記述する指標を作成する
 - 複数銘柄からなる株価の時系列データから、市場全体の変動を記述する総合指標を作成する

例：都道府県別の生活環境

県名	地方名	昼夜人口比	年少人口比	老年人口比	人口増減率	粗出生率	粗死亡率	婚姻率	離婚率	高校数／人
北海道	北海道	100.0	11.7	26.0	-0.47	7.09	10.63	4.86	2.12	192.6
青森県	東北	100.0	12.1	27.0	-0.95	6.79	12.81	4.33	1.78	194.3
岩手県	東北	99.7	12.4	27.9	-0.84	7.12	12.33	4.32	1.52	195.1
宮城県	東北	100.2	13.0	22.9	-0.09	8.05	9.51	5.30	1.70	146.3
秋田県	東北	99.9	11.1	30.7	-1.12	6.16	13.98	3.78	1.41	186.7
山形県	東北	99.8	12.6	28.3	-0.78	7.13	12.81	4.24	1.46	178.0
福島県	東北	99.6	12.9	26.1	-1.41	7.02	11.94	4.73	1.64	171.3
茨城県	関東	97.2	13.2	23.8	-0.51	7.78	10.20	4.92	1.79	138.3
栃木県	関東	99.1	13.2	23.2	-0.40	8.02	10.43	5.13	1.85	136.7
群馬県	関東	99.9	13.4	24.9	-0.45	7.49	10.63	4.64	1.77	135.4
埼玉県	関東	88.6	13.0	22.0	0.07	7.90	8.20	5.10	1.86	96.8
千葉県	関東	89.5	12.8	23.2	-0.31	7.89	8.59	5.19	1.86	110.4
東京都	関東	118.4	11.3	21.3	0.26	8.12	8.25	6.75	1.91	144.6
神奈川県	関東	91.2	13.0	21.5	0.10	8.32	7.94	5.68	1.85	98.4
新潟県	中部	100.0	12.5	27.2	-0.64	7.45	11.97	4.35	1.37	148.9
富山県	中部	99.8	12.7	27.6	-0.55	7.28	11.79	4.50	1.43	171.3
石川県	中部	100.2	13.4	25.0	-0.26	8.21	10.51	4.91	1.52	163.0
福井県	中部	100.1	13.7	26.0	-0.50	8.40	11.01	4.55	1.55	158.2
山梨県	中部	99.0	12.9	25.6	-0.58	7.44	11.21	4.60	1.87	158.1
長野県	中部	99.9	13.5	27.4	-0.47	7.81	11.48	4.67	1.66	159.3
岐阜県	中部	96.0	13.7	25.2	-0.48	8.00	10.45	4.62	1.60	127.8
静岡県	中部	99.9	13.4	24.9	-0.37	8.25	10.23	5.17	1.84	131.5
愛知県	中部	101.5	14.2	21.4	0.15	9.14	8.26	5.75	1.82	103.0
三重県	関西	98.1	13.5	25.3	-0.38	8.00	10.44	4.89	1.76	128.6
滋賀県	関西	96.6	14.8	21.6	0.07	9.35	8.64	5.22	1.66	134.8
京都府	関西	101.2	12.6	24.7	-0.27	7.66	9.68	5.02	1.77	145.4
大阪府	関西	104.7	13.0	23.7	-0.06	8.24	9.09	5.43	2.12	107.8
兵庫県	関西	95.7	13.5	24.3	-0.20	8.34	9.63	5.07	1.84	131.6
奈良県	関西	89.9	12.9	25.5	-0.43	7.60	9.82	4.48	1.72	122.7
和歌山県	関西	98.1	12.5	28.4	-0.70	7.51	12.59	4.72	1.98	168.2
鳥取県	中国	100.0	13.2	27.2	-0.51	8.20	12.15	4.74	1.83	171.8
島根県	中国	100.0	12.7	30.0	-0.70	7.90	13.46	4.40	1.43	228.2
岡山県	中国	99.9	13.5	26.2	-0.26	8.41	10.94	4.94	1.82	157.5
広島県	中国	100.3	13.5	25.3	-0.25	8.72	10.28	5.15	1.78	155.4
山口県	中国	99.5	12.6	29.2	-0.76	7.55	12.74	4.58	1.67	206.3
徳島県	四国	99.7	12.2	28.0	-0.51	7.40	12.60	4.34	1.62	175.1
香川県	四国	100.2	13.2	27.1	-0.30	8.25	11.50	4.84	1.91	149.4
愛媛県	四国	100.1	12.8	27.8	-0.56	7.87	12.17	4.51	1.79	156.8
高知県	四国	99.9	11.9	30.1	-0.79	7.00	13.49	4.33	1.87	208.4
福岡県	九州	100.1	13.5	23.3	0.12	9.01	9.63	5.50	2.07	112.3
佐賀県	九州	100.2	14.4	25.3	-0.47	8.83	11.48	4.75	1.74	159.8
長崎県	九州	99.8	13.4	27.0	-0.64	8.33	11.92	4.50	1.74	169.3
熊本県	九州	99.6	13.7	26.5	-0.33	8.85	11.38	4.96	1.87	144.2
大分県	九州	100.0	12.9	27.6	-0.50	8.14	11.86	4.77	1.85	181.5
宮崎県	九州	100.0	13.8	26.7	-0.44	8.75	11.59	5.03	2.15	146.6
鹿児島県	九州	99.9	13.6	27.0	-0.53	8.78	12.59	4.78	1.84	166.9
沖縄県	九州	100.0	17.6	17.7	0.57	12.12	7.54	6.28	2.58	122.1

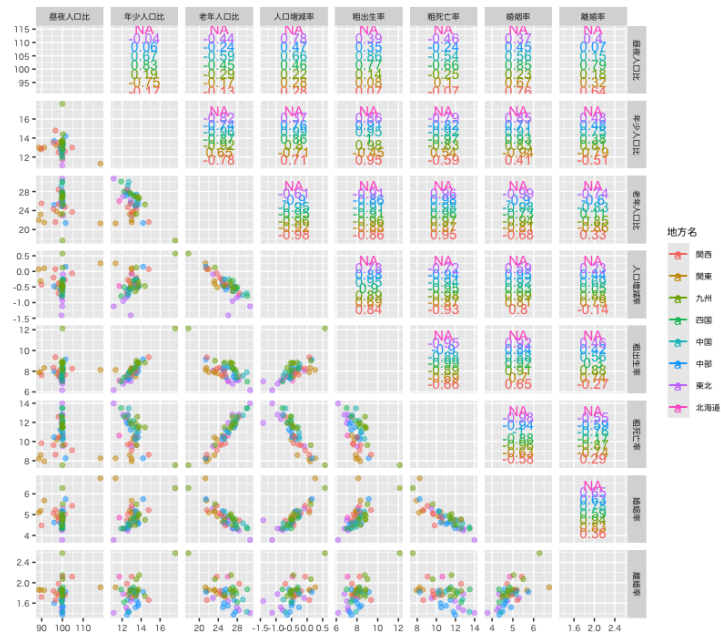


Figure 6: 都道府県別の人口動態

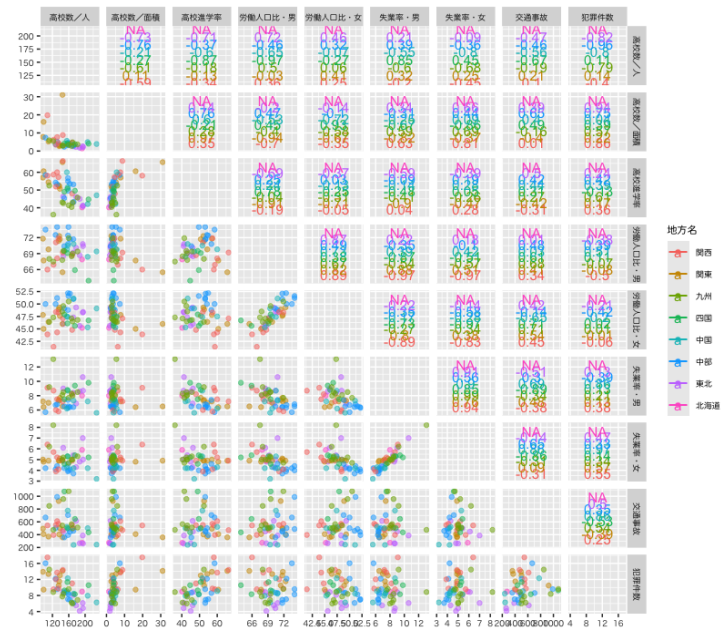


Figure 7: 都道府県別の教育・労働環境

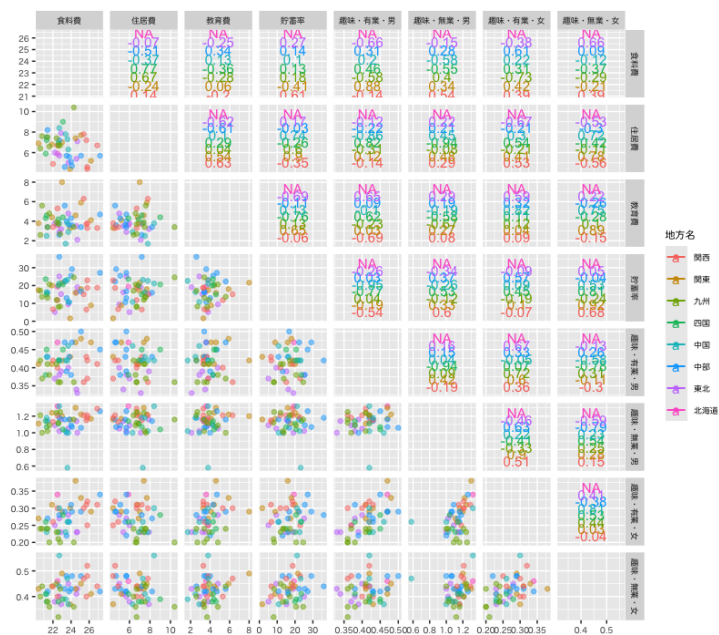


Figure 8: 都道府県別の貯蓄・余暇

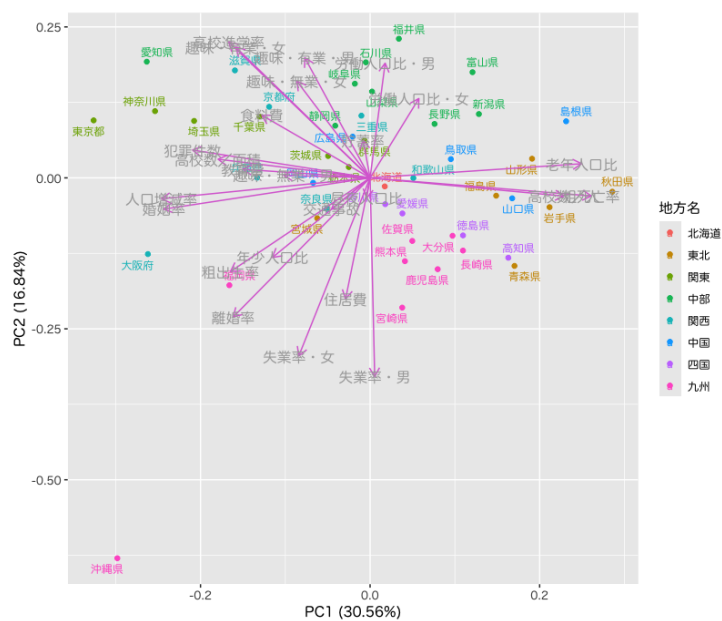


Figure 9: 都道府県別の生活環境の主成分分析

判別分析の考え方

- ある個体が複数のクラスのいずれかに属するとき、その個体の特徴量からどのクラスに属するかを予測するモデルを構築する
- 分析の事例
 - 食道がんを患っている人とそうでない人を、年齢・飲酒量・喫煙度から判別する
 - 銀行が融資判断をするために、企業の財務データから、その企業が期間内に債務不履行となるか否かを予測する

例：乳癌患者の生研検査

V1	V2	V3	V4	V5	V6	V7	V8	V9	class
5	1	1	1	2	1	3	1	1	benign
5	4	4	5	7	10	3	2	1	benign
3	1	1	1	2	2	3	1	1	benign
6	8	8	1	3	4	3	7	1	benign
4	1	1	3	2	1	3	1	1	benign
8	10	10	8	7	10	9	7	1	malignant
1	1	1	1	2	10	3	1	1	benign
2	1	2	1	2	1	3	1	1	benign
2	1	1	1	2	1	1	1	5	benign
4	2	1	1	2	1	2	1	1	benign
1	1	1	1	1	1	3	1	1	benign
2	1	1	1	2	1	2	1	1	benign
5	3	3	3	2	3	4	4	1	malignant
1	1	1	1	2	3	3	1	1	benign
8	7	5	10	7	9	5	5	4	malignant

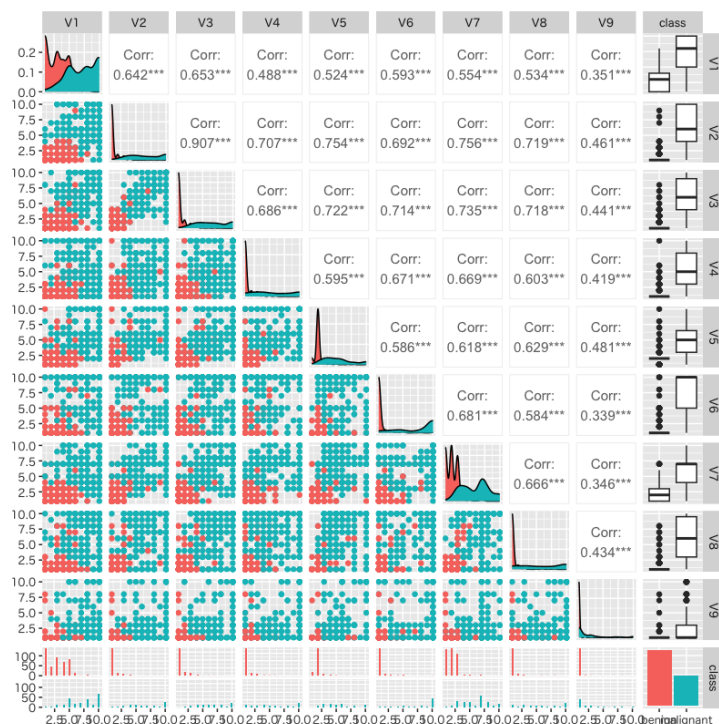


Figure 10: 乳癌患者 (良性・悪性) の生研検査の散布図

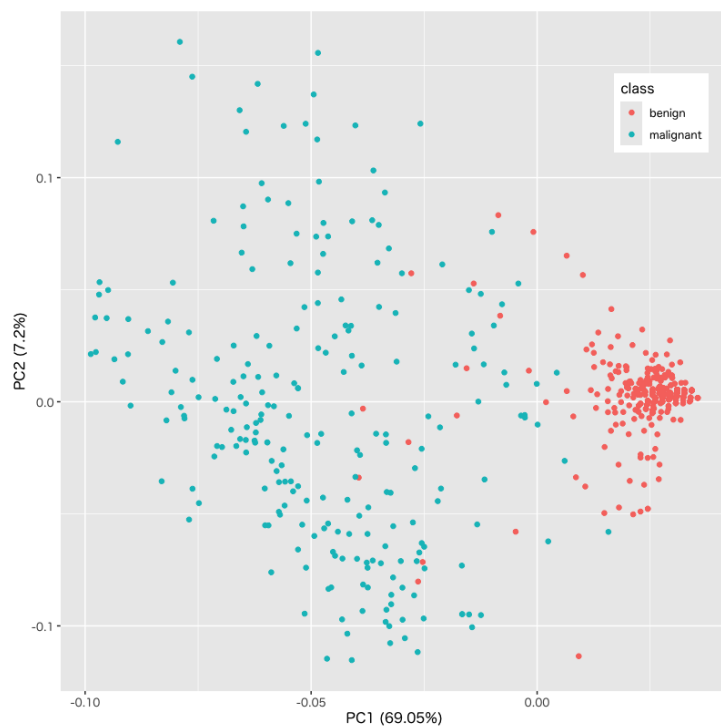


Figure 11: 生研検査の主成分分析

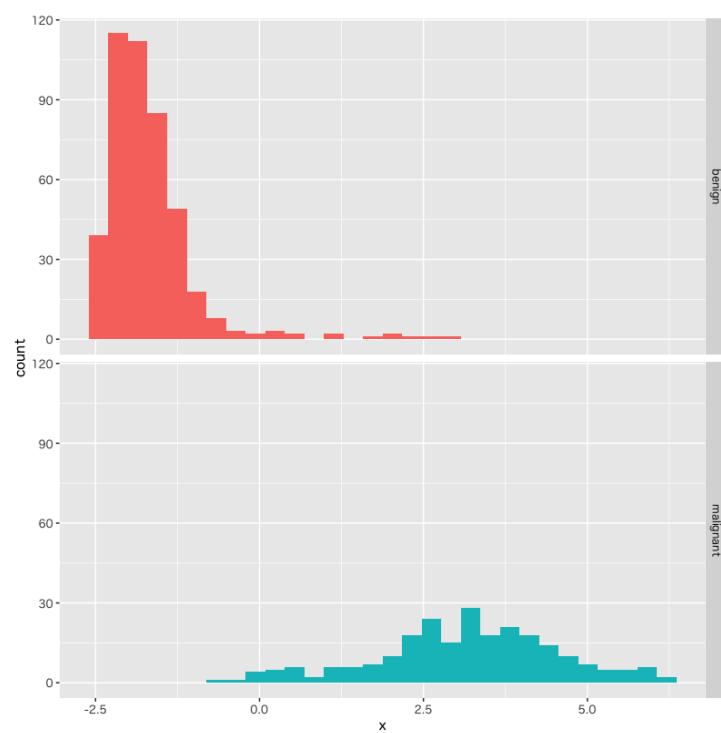


Figure 12: 生研検査による乳癌患者の判別分析

クラスタ分析の考え方

- ・ 特徴量の違いに着目して、妥当な個体のグループ(クラスタ)を構成する
 - 階層的な方法：系統樹を作成する
 - 非階層的な方法：グループの代表値を推定する
- ・ 分析の事例
 - 映画に関するアンケート調査から潜在的なジャンル(グループ)を抽出する
 - 顧客の購買履歴から、嗜好の異なる顧客グループに分類し、グループごとの販売戦略を立てる

例：好きなおむすびの具に関するアンケート

県名	梅	鮭	昆布	鰹	明太子	鰯子	ツナ	その他
北海道	13.86	27.94	5.58	5.26	9.26	15.06	11.61	11.39
青森県	14.93	30.79	7.01	2.43	10.36	11.58	11.58	11.28
岩手県	17.91	23.13	5.22	3.35	17.91	10.07	10.44	11.94
宮城県	15.16	29.50	10.00	1.66	14.83	8.83	12.83	7.16
秋田県	10.63	31.38	5.31	3.19	14.89	13.29	10.63	10.63
山形県	16.58	20.27	8.29	1.38	18.89	10.13	12.90	11.52
福島県	12.37	21.99	8.93	3.43	16.49	9.62	19.24	7.90
茨城県	15.42	26.49	7.98	2.54	18.33	11.79	11.79	5.62
栃木県	16.61	27.04	10.70	1.97	16.90	9.29	12.67	4.78
群馬県	14.24	22.53	6.21	1.81	20.20	15.28	13.73	5.95
埼玉県	13.91	27.17	8.03	3.57	19.36	10.17	13.58	4.18
千葉県	14.96	28.49	7.10	3.67	19.07	10.53	10.97	5.17
東京都	14.07	28.03	8.28	3.28	17.80	11.74	11.20	5.56
神奈川県	15.05	28.06	9.61	3.47	16.10	10.67	11.49	5.51
新潟県	21.06	23.03	3.37	1.12	19.66	15.16	7.02	9.55
富山県	19.41	21.84	14.56	4.85	13.59	7.76	11.16	6.79
石川県	23.05	20.00	19.66	4.06	13.89	7.11	10.16	2.03
福井県	20.58	20.58	18.62	4.90	15.68	6.86	9.80	2.94
山梨県	14.11	20.58	9.41	2.35	20.58	12.35	18.23	2.35
長野県	18.64	21.30	9.68	2.66	20.33	13.80	10.16	3.38
岐阜県	16.52	20.55	11.44	4.44	15.88	8.68	15.46	6.99
静岡県	15.96	27.71	11.86	4.98	14.30	9.20	10.75	5.21
愛知県	15.38	27.62	10.87	3.10	14.77	8.23	13.64	6.35
三重県	13.51	29.50	11.26	5.18	14.63	7.88	10.58	7.43
滋賀県	14.96	23.12	12.92	6.12	18.70	7.14	12.24	4.76
京都府	18.99	26.62	12.66	5.68	15.09	8.44	8.92	3.57
大阪府	17.19	25.33	13.71	6.87	14.54	8.22	10.49	3.61
兵庫県	17.93	23.82	13.85	5.74	15.23	7.34	12.04	4.01
奈良県	16.45	24.83	11.61	8.38	14.83	9.35	10.64	3.87
和歌山県	16.22	20.61	15.78	7.89	12.28	9.21	14.47	3.50
鳥取県	18.64	16.10	10.16	2.54	22.03	11.86	13.55	5.08
島根県	20.33	18.64	15.25	6.77	15.25	6.77	12.71	4.23
岡山県	17.30	18.20	12.80	4.49	18.20	9.43	14.38	5.16
広島県	22.03	21.12	17.17	6.38	17.17	6.23	7.44	2.43
山口県	22.22	17.20	16.84	2.86	20.07	5.37	9.67	5.73
徳島県	22.06	20.68	17.24	7.58	13.79	5.51	11.03	2.06
香川県	17.75	22.42	12.61	4.67	13.08	8.41	14.95	6.07
愛媛県	21.72	25.29	11.60	5.95	15.17	5.95	12.50	1.78
高知県	11.72	33.79	17.93	3.44	9.65	10.34	9.65	3.44
福岡県	15.62	22.59	11.73	4.94	22.77	5.11	11.65	5.56
佐賀県	17.98	19.42	5.75	3.59	12.94	9.35	23.74	7.19
長崎県	23.36	22.13	11.88	7.78	16.39	3.68	8.60	6.14
熊本県	21.57	14.72	13.69	7.19	19.52	4.79	12.67	5.82
大分県	14.00	27.53	11.11	4.83	20.77	3.38	9.66	8.69
宮崎県	20.88	22.15	10.75	3.79	18.35	11.39	7.59	5.06
鹿児島県	17.59	24.53	9.25	4.16	19.44	6.01	14.81	4.16
沖縄県	19.59	22.85	7.34	2.04	15.10	4.48	11.42	17.14

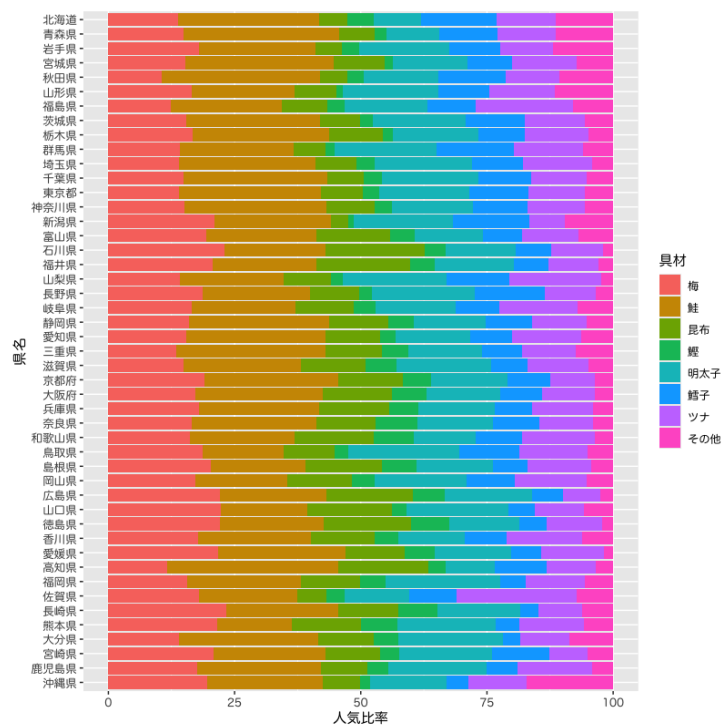


Figure 13: 都道府県別の好きなおむすびの具の集計結果

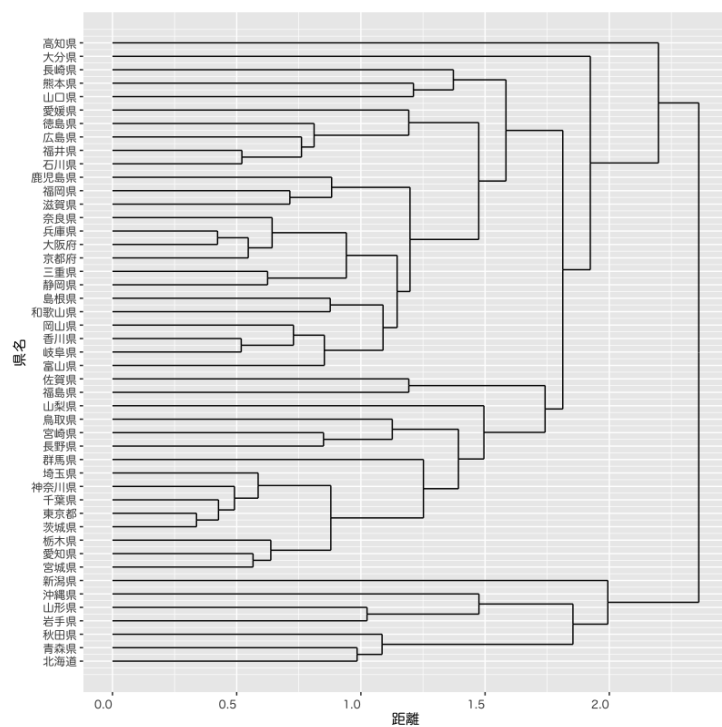


Figure 14: アンケート結果にもとづく県のクラスタ分析

時系列解析の考え方

- 時間とともに変化する現象を記述するために、未来の値を過去の値で近似する式を構成する
 - 自己回帰 (AR モデル) : 過去の影響の記述
 - 移動平均 (MA モデル) : 記憶のある不確定性
- 分析の事例
 - 市町村の過去の年齢別の人口変動から将来の人口比率の推移を予測する
 - 食品・飲料の季節ごとの販売履歴から、将来の需要量を予測して生産計画を立てる

例：米国航空機旅客量の変遷

Year	1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月
1949	112	118	132	129	121	135	148	148	136	119	104	118
1950	115	126	141	135	125	149	170	170	158	133	114	140
1951	145	150	178	163	172	178	199	199	184	162	146	166
1952	171	180	193	181	183	218	230	242	209	191	172	194
1953	196	196	236	235	229	243	264	272	237	211	180	201
1954	204	188	235	227	234	264	302	293	259	229	203	229
1955	242	233	267	269	270	315	364	347	312	274	237	278
1956	284	277	317	313	318	374	413	405	355	306	271	306
1957	315	301	356	348	355	422	465	467	404	347	305	336
1958	340	318	362	348	363	435	491	505	404	359	310	337
1959	360	342	406	396	420	472	548	559	463	407	362	405
1960	417	391	419	461	472	535	622	606	508	461	390	432

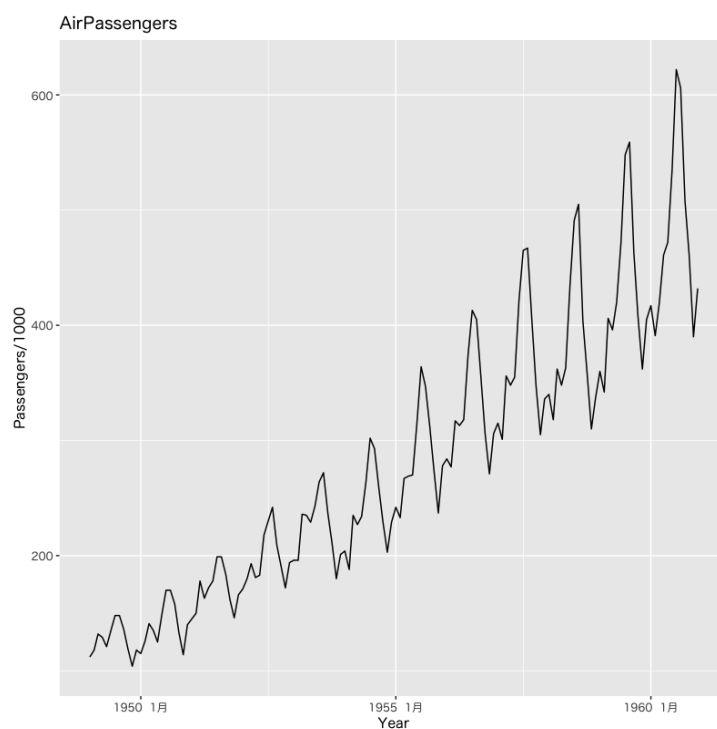


Figure 15: 旅客量の変遷

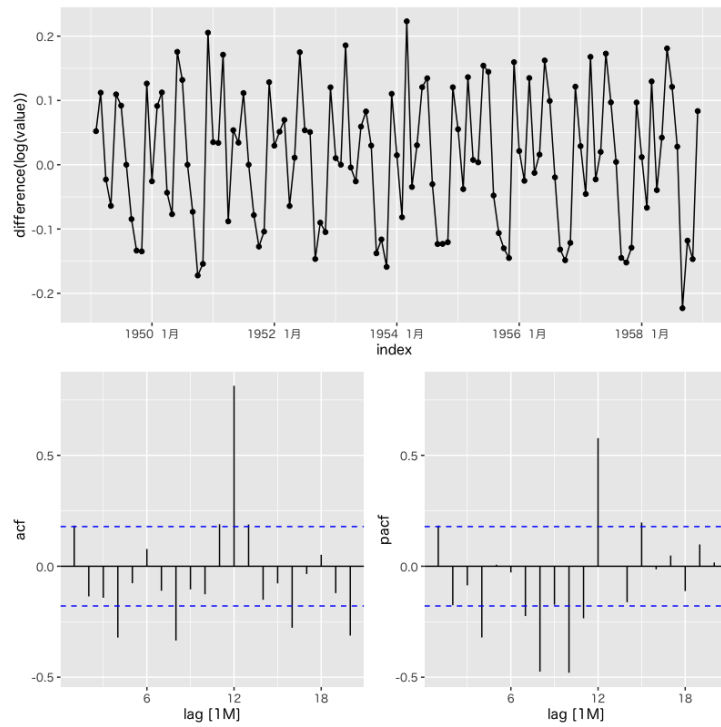


Figure 16: 階差時系列の自己相関分析

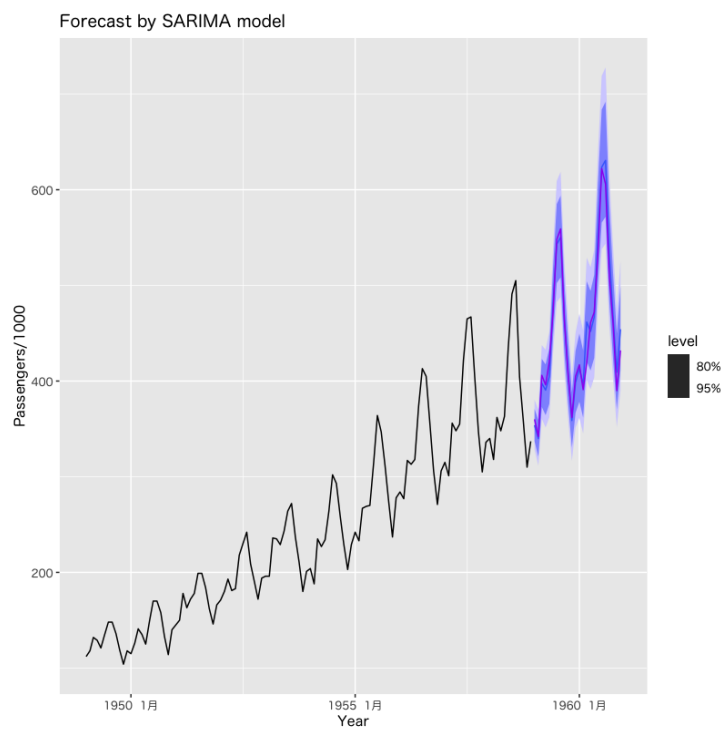


Figure 17: SARIMA モデルによる旅客量の予測

RStudio の構成

起動画面

- 以下 **RStudio** を用いて説明
- 複数の **タブ** (tab; つまみ) を含む 4 つの **ペイン** (pane; 枠) が立ち上がる
 - 左上: エディタ・表など (開いていない場合もある)
 - 左下: コンソール・ターミナルなど
 - 右上: 作業環境内の変数・コマンド履歴など
 - 右下: パッケージ・グラフィックス・ヘルプなど
- ペインの配置や数は個別に設定することが可能
 - メニュー: **Tools > Global Options** で設定

コンソール (左下ペイン)

- R 言語で記述されたコマンドを入力
 - 例えば以下のような計算を行うことができる

```
#' 一般的な数式を入力すれば計算機として使える
#' "#" 以降はコメントとして無視される
1 + 2 + 3 + 4 # 空白は無視される
sin(pi/3) / cos(pi/3) # tan(pi/3) になるはず
```

- コンソール上で終了を指示する以下のコマンドを入力すれば R を終了させることができる

```
#' R の終了には q() または quit() を用いる
q()
```

- * メニューの **Quit RStudio** も利用可能
- * 終了できない場合は OS の機能で強制終了する必要がある

エディタ (左上ペイン)

- コマンドを記述したファイル进行操作
- 一連のコマンドをまとめたり修正しながら実行
 - コンソールに入力したコマンドは直ちに実行されてしまう
 - コマンドを実行順に記述したファイル **R Script** を作成
 - ファイル (の一部) を実行
 - ファイルを保存
- 同一ファイル内でプログラムと文書の記述も可能
 - **Quarto** (本講義でも利用)
 - R Markdown

ヘルプ (右下ペイン)

- 各関数の詳細を記述したヘルプが参照可能
 - 機能, 引数名, 引数の既定値, 実行例など
- ヘルプ内の検索はその下の **Find in Topic** で可能
- コンソール内では関数 `help()` や `?` などを利用

パッケージ (右下ペイン)

- パッケージを用いて機能を拡張
- RStudio でのインストール手順
 - 右下ペイン **Package** タブをクリック
 - 左上の **Install** をクリック
 - パッケージ名を入力し **Install** をクリック
- 利用可能なパッケージの情報は右下ペイン **Package** タブで確認できる

作業ディレクトリ

- プログラムが実行されるディレクトリ (フォルダ)
- 作業ディレクトリにあるファイルの読み書きはパスを指定する必要がある
- 現在の作業ディレクトリは **Console** タブで確認
- メニュー: **Session > Set Working Directory** で指定
 - 読み込んだファイルの場所を選択
 - **Files** タブ (右下ペイン) の場所を選択 (**More** から選択可)
 - ディレクトリを直接選択

プロジェクト

- 作業環境をまとめて設定・保存する機能
 - 作成したプロジェクトは **Project** ボタン (右上) から選択可能
 - いつでもプロジェクトを中断可能
 - プロジェクト毎に履歴や変数を保存可能
 - 複数のプロジェクトを定義可能
- 一般的なプロジェクトの作成手順
 - **Project** ボタンから **New Project** を選択
 - **Create Project** ダイアログで **New Directory** を選択
 - * 既にあるディレクトリを用いる場合は **Existing Directory**
 - * Github などを利用する場合は **Version Control**
 - **Project Type** ダイアログで適切なタイプを選択
 - * 講義では **Quarto Project** を利用
 - **Directory Name** とその親ディレクトリを指定

終了時の注意

- 終了時にコンソールに以下のメッセージ, または同様の内容のダイアログボックスが表示される場合がある

```
> q()  
Save workspace image? [y/n/c]:
```

- 作業で使った変数などをセーブするか尋ねている
 - y を入力: セーブする (yes の略)
 - n を入力: セーブしない (no の略)
 - c を入力: R の終了をキャンセルする (cancel の略)
- セーブした場合次回起動時に読み込まれる

Quarto の利用法

Quarto ファイルの作成

- 左上の + ボタン (または “File/New File”) を開く
- **Quarto Document** を選択する
- ダイアログに必要な情報を記入する
 - 文書の “Title/Author” の設定
 - フォーマットの選択 (HTML/PDF/Word)
 - * 講義では HTML を利用
- **Create** をクリックするとファイルが作成される
 - “Create” は雛型の文書付き
 - “Create Empty Document” は YAML ヘッダーのみ
 - * 慣れたらこちらで

エディタの設定

- Quarto ファイルは左上のペインに表示される
- 左上の “Source/Visual” ボタンでモードが変わる
 - **Source**: 装飾なしのテキストエディタ
 - **Visual**: Markdown 記法に従った装飾ありの表示
- 各モードでの注意
 - Visual モードは GUI を利用して装飾を変更する
 - History 機能は現在 Source モードでしか機能しない

コードの入力と実行

- コードを入力するにはチャンクを作成する
 - 入力したい箇所にカーソルを移動
 - 上枠の +C (緑) ボタン (または “Code/Insert Chunk”) をクリック
- 歯車 ボタンで実行時の出力先を設定する
 - **Chunk Output Inline**: エディタ内に出力
 - **Chunk Output in Console**: コンソール内に出力
 - * 講義ではコンソールを利用
- 実行法はいくつか用意されている (**Run** ボタン参照)
 - ► (緑色): チャンク内を実行
 - ▽ (灰色): そのチャンク以前を実行
 - * カーソル位置のみ実行することも可能
 - * ショートカットキーも用意されている

出版

- 全体をまとめて整形することができる
- 歯車 ボタンで出力時の挙動を設定する
 - **Preview in Window** : ブラウザに表示
 - **Preview in Viewer Pane** : RStudio の Viewer ペインに表示
 - * 講義では基本的に Viewer を利用
 - * Viewer の上の左から 2 番目のボタンでブラウザに送ることは可能
- Quarto と同じ場所にファイルは保存される
 - フォーマットによって拡張子が異なる
 - 付加的なファイルは“ファイル名 + files”に保存される
 - 単一のファイルにするには YAML ヘッダーで以下を宣言する

```
---
format:
  html:
    self-contained: true
---
```

ファイルを保存する

- 以下のいずれかで保存することができる
 - 左上のディスクのマークをクリック
 - **File** から **Save** を選択 (Ctrl/Command+S)
- ファイル作成に関する注意
 - 保存する時にファイル名の入力求められる
 - * **Quarto** の拡張子は通常 **.qmd** を利用
 - 文書はヘッダを付けてわかりやすく構造しておくが良い
 - チャンク内の # 以降の文字列は実行されないのでコメントとして有用

基本的な使い方

計算をする

- 四則演算や数学関数は直感的な文法で計算可能

Table 1: 基本的な演算と関数

加減乗除	+, -, *, /
冪乗	^ または **
三角関数	sin(), cos(), tan()
逆三角関数	asin(), acos(), atan()
指数関数	exp()
対数関数	log(), log10(), log2()
双曲線関数	sinh(), cosh(), tanh()
平方根	sqrt()
絶対値	abs()

電卓として使う

- コンソール上での計算例

```
#' 与えられた式の計算をコンソール上で実行する
#' 1 x 2 + 3^2 の計算
1 * 2 + 3^2
#' sin(2π) の計算
sin(2*pi)
#' √ 2 + |-0.6| の計算
sqrt(2) + abs(-0.6)
```

```
[1] 11
[1] -2.449294e-16
[1] 2.014214
```

- 入力内容は右上の **History** タブで確認可能

関数

- 関数の取り扱いは一般的な計算機言語とほぼ同様
 - 関数は引数とその値を指定して実行 (引数がない場合もある)
 - 引数名は順序を守れば省略可能
- 関数の呼び出し方 (関数名を **f** とする)

```
f(arg1 = value1, arg2 = value2) # 擬似コード
#' arg1, arg2 は引数の名前, value1, value2 は引数に渡す値を表す
f(value1, value2) # 上と同値. 順序に注意
```

- 擬似コード = 実行しても動かないコード

- 実装されている関数の使い方は **help(関数名)** または **?関数名** でヘルプが表示される

関数の実行例

- 正弦関数の計算

```
#' 正弦関数 (引数が 1 つ) の計算例
sin(x = pi/2)
sin(pi/2) # 引数名は省略でき, 前の行とこの行は同じ結果になる
```

- 対数関数の計算

```
#' 対数関数 (引数が 2 つ) の計算例
#' 以下は擬似コード. a, b を適当な数値に置き換えて実行しなさい
log(a, b) # 底を b とする a の対数
log(x = a, base = b) # 上と同値
log(base = b, x = a) # 上と同値 (引数名があれば順序は自由に変えられる)
log(b, a) # = log(x=b, base=a) (引数名がなければ規定の順序で解釈される)
log(a) # 自然対数 = log(a, base=exp(1))
```

ヘルプ機能

- 各関数の詳細を記述したヘルプが用意されている
 - Description (機能の概要)
 - Usage (関数の呼び出し方)
 - Arguments (関数の引数)

- Value (関数の返り値)
- Examples (実行例)
- ヘルプに関連する関数
 - `help()` (使い方や例の表示)
 - `example()` (例を実際に実行してくれる)
 - `help.search()` (キーワード検索)
- 右下ペイン **Help** タブの利用
 - 右上にある検索窓でヘルプを参照可能
 - 左上にある検索窓はヘルプ内を検索可能

ヘルプ機能の利用例

- ヘルプの使い方

```
#' 関数 log() に関するヘルプの例
help(log) # Help タブに結果は表示される
?log # 上と同値
example(log) # ヘルプ内の例を実行
help.search("log") # "log"に関連する項目は?
??"log" # 上と同値
```

- 右下ペインの **Help** タブに表示される

データ型

- R ではさまざまな数値を扱うことができる
 - 実数および複素数 (指数表記にも対応)
 - 無限大や不定な数など特殊なものにも対応

Table 2: 代表的なデータ型

型の名称	役割	例
numeric	(広義の) 実数を表す	1, pi, NaN
complex	複素数を表す	1i, 3-4i
character	文字列を表す	"foo", "Hello World!"
logical	論理値 (真偽) を表す	TRUE, FALSE, 3<4, NA

オブジェクト

- 変数, 関数, 計算結果などを **オブジェクト** と呼ぶ
- 文字列を変数名としてオブジェクトを保持することができる
- オブジェクトの内容を別のオブジェクトに代入することができる
- 保持しているオブジェクトの情報は右上ペインの **Environment** タブで確認できる

オブジェクトの代入の例

- 代入操作の例

```
#' 数値を変数 foo に代入する
(foo <- 3) # foo <- 3; print(foo) と等価
#' 変数 foo を用いて計算し, 結果を bar に代入する
bar <- sin(2/3*pi) + cos(foo * pi/4) # 計算結果は表示されない
#' 変数 bar の内容を表示する
```

```
print(bar)
```

```
[1] 3  
[1] 0.1589186
```

- 計算結果や良く使う文字列の保存に利用できる
- 変数名は自由に決められるが、予約語 `c`, `q`, `t`, `C`, `D`, `F`, `I`, `T` には注意が必要

自作関数

- 他の言語と同様に R でも関数を定義できる
- 関数の定義には関数 `function()` を利用する

```
#' 関数 function() の使い方 (擬似コード)  
関数名 <- function(引数){ # 計算ブロックの開始  
  ## このブロック内に必要な手続きを記述する。複数行に渡って構わない  
  return(返値) # 計算結果を明示的に示す  
} # ブロックの終了
```

自作関数の例

- 縦と横の長さを与えて長方形の面積を計算

```
#' 縦の長さ a, 横の長さ b (既定値は 1) の長方形の面積  
foo <- function(a, b = 1){  
  out <- a * b  
  return(out) # 計算結果を外に返却  
}  
#' 実行例  
foo(2, 3) # foo(a = 2, b = 3) と同義  
foo(2)   # foo(a = 2, b = 1) と同義
```

```
[1] 6  
[1] 2
```

- 無名関数 (anonymous function) の作り方

```
#' 変数や関数を定義して計算する方法  
(x <- 1:10/10)  
foo <- function(x) sin(x)/x # 式の計算結果を返却 (return を省略可)  
foo(x)  
#' 変数や関数を定義せずに計算する方法  
(function(x) sin(x)/x)(1:10/10)  
(\ (x) sin(x)/x)(1:10/10) # R 4.1 以降の短縮表現
```

```
[1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0  
[1] 0.9983342 0.9933467 0.9850674 0.9735459 0.9588511 0.9410708 0.9203110 0.8966951  
[9] 0.8703632 0.8414710  
[1] 0.9983342 0.9933467 0.9850674 0.9735459 0.9588511 0.9410708 0.9203110 0.8966951  
[9] 0.8703632 0.8414710  
[1] 0.9983342 0.9933467 0.9850674 0.9735459 0.9588511 0.9410708 0.9203110 0.8966951  
[9] 0.8703632 0.8414710
```

- オブジェクト (変数や関数) を作成する必要がある場合に有用

- 練習

- R の関数 `integrate()` についてヘルプを調べなさい.
- 以下の関数の定積分を求めなさい.

$$f(x) = \frac{1}{1+x^2}, \quad x \in [0, 1]$$

$$g(x) = \exp\left(-\frac{x^2}{2}\right), \quad x \in \mathbb{R}$$

- 関数 `integrate()` を利用して定積分を計算

```
#' 関数 1/(1+x^2) を区間 [0,1] で積分 (関数オブジェクトを渡す書き方)
f <- function(x) 1/(1+x^2) # 関数を定義
integrate(f, 0, 1) # pi/4
#' 関数 e^(-x^2/2) を実軸全体で積分 (Gauss 積分) (関数を定義しない書き方)
integrate(\(x)exp(-x^2/2), -Inf, Inf) # sqrt(2*pi)
```

```
0.7853982 with absolute error < 8.7e-15
2.506628 with absolute error < 0.00023
```

次回の予定

- データの取り扱い
- 描画の基礎
- 確率シミュレーション