CoursewareHubを活用した情報教育の実践について

2020年9月25日

室蘭工業大学
桑田喜隆

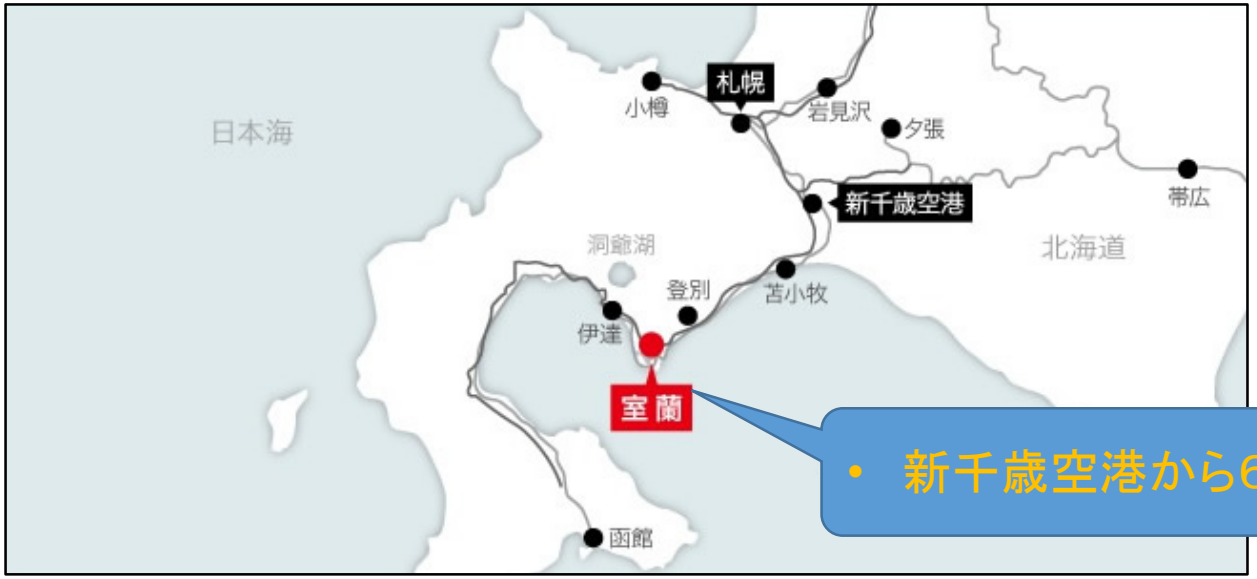
自己紹介

室蘭工業大学
情報教育センター　センター長
桑田喜隆（くわたよしたか）

1986年群馬大学電子工学科修了.その後,NTTデータで人工知能,グループウエア,GIS,ロボカップレスキュー,分散処理システム等の研究開発に従事. 2014 年より室蘭工業大学で情報教育を担当.クラウドコンピューティングの研究を実施中.

室蘭工業大学の概要

- 工学系単科大学
- 学部の改組（工学部から理工学部へ、2019年度）
- 学生数：3400人
- 教職員数：300人



- 新千歳空港から60分

Jupyter Notebookで始めるプログラミング

はじめてプログラミングを学習する人のための入門書.
Jupyter Notebook上でプログラミング言語「Python」を使って対話的に学ぶことができる.

桑田 喜隆, 小川 祐紀雄, 早坂 成人, 石坂 徹

ISBN-10 : 4780608589
出版社 : 学術図書出版社
ISBN-13 : 978-4780608588

9月30日発売予定

概要

- 背景と仮説
- 模擬授業
- 2019年度授業
- 授業の分析
- まとめと今後の課題

背景と仮説

はじめに

- プログラミング教育が注目される
  - 新学習指導要領（情報教育・ICT活用教育関係）２０１７

  - 小・中・高等学校共通のポイント
  - 情報活用能力＝学習の基盤となる資質・能力
  - プログラミング的思考

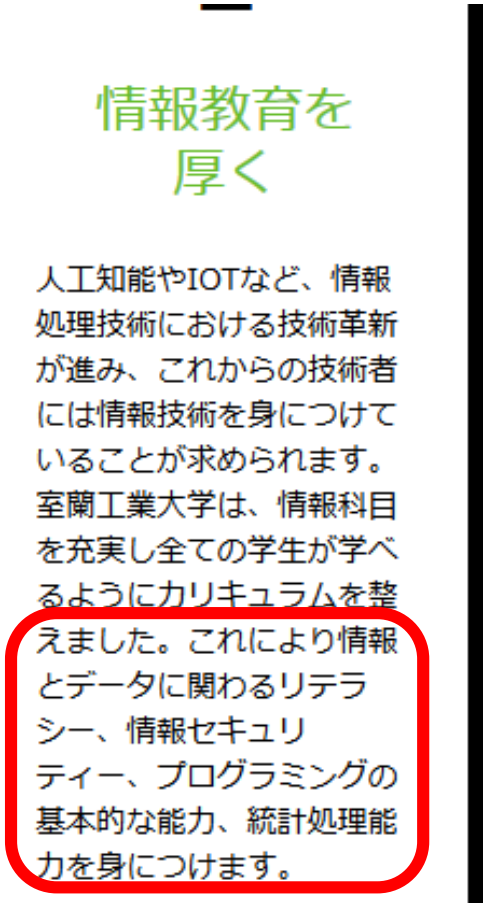
- 現状（工学系大学）
  - プログラミング未経験者
  - ２割程度が「苦手」
  - 専門科目の履修に必要なコースもあるが、言語はバラバラ

- 課題
  - 大学の一般教養として何を身につけるべきか？

情報教育の強化

- 2019年度より工学部から理工学部へ改組

室蘭工業大学では、大学が改組され、2019年度より理工学部全学科でプログラミングが必修科目となる。



プログラミング入門
(2019後期より開講)

- シラバス

対象学年	1年
授業科目区分	学科共通科目
必修・選択	必修
授業方法	講義、演習
単位数	2
到達度目標	本講義では、全コースの学生を対象に、プログラミングに必要な概念を理解し、基礎的なプログラムを作成することができるようになることを目標とする。

演習をめぐる仮説

【仮説１】座学に加え演習を実施することで学生の理解度が向上する

演習中の試行錯誤でプログラミングに関する理解が進み，知識が深まる.

【仮説２】演習が進むほど学生の理解度が低下する

概念を理解しないまま次の演習に進んでしまい，次の演習ではますます分からなくなるというリスクがある.

【仮説３】理解度に応じた教育で，理解度の低下を防ぐことができる

学生の理解度を把握し適切な助言や指導を行うことが重要であると考える.

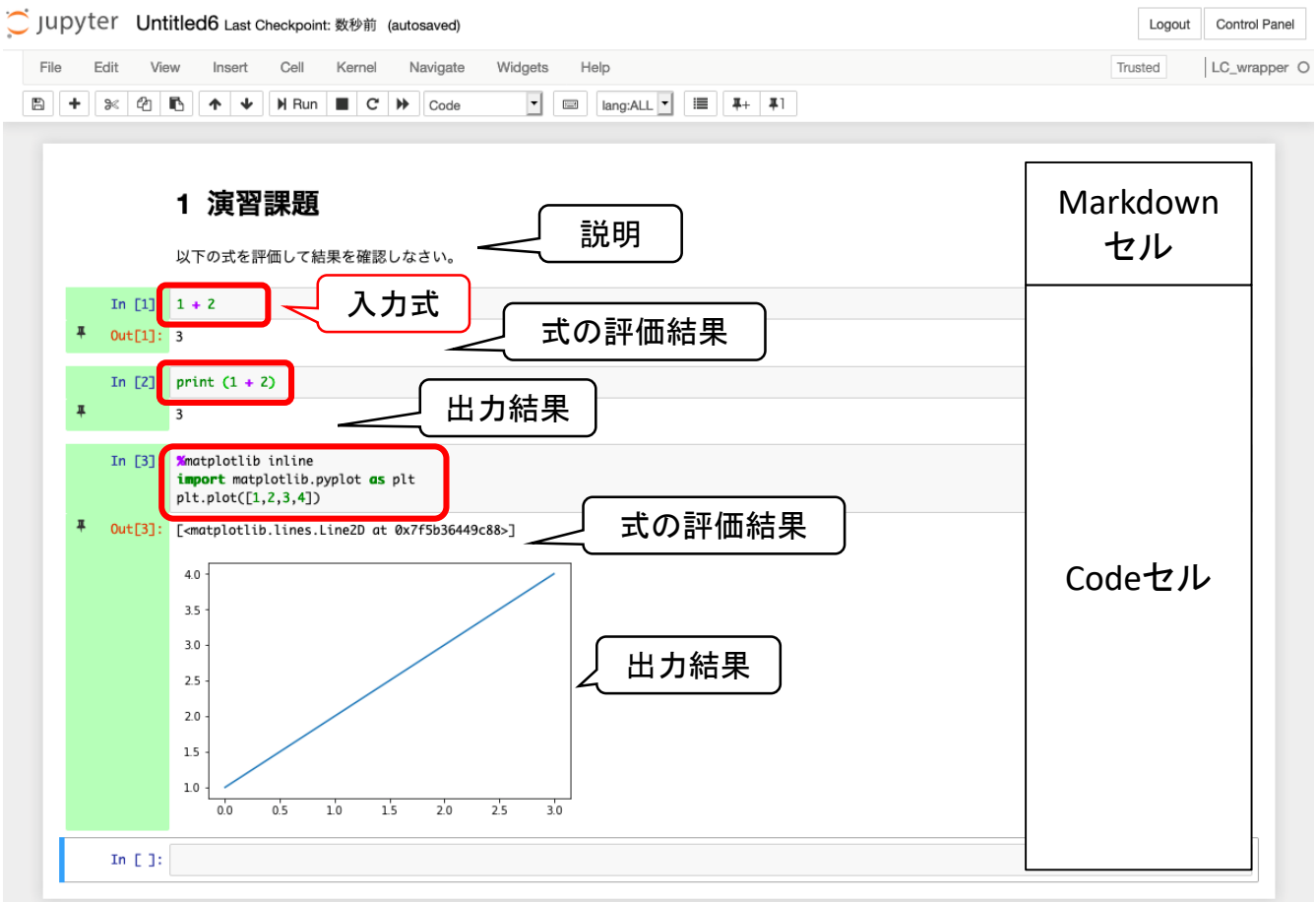
Jupyter Notebookとは

Webベースの対話的な計算機環境
科学技術計算やデータ解析，データの可視化で利用が広がっている.

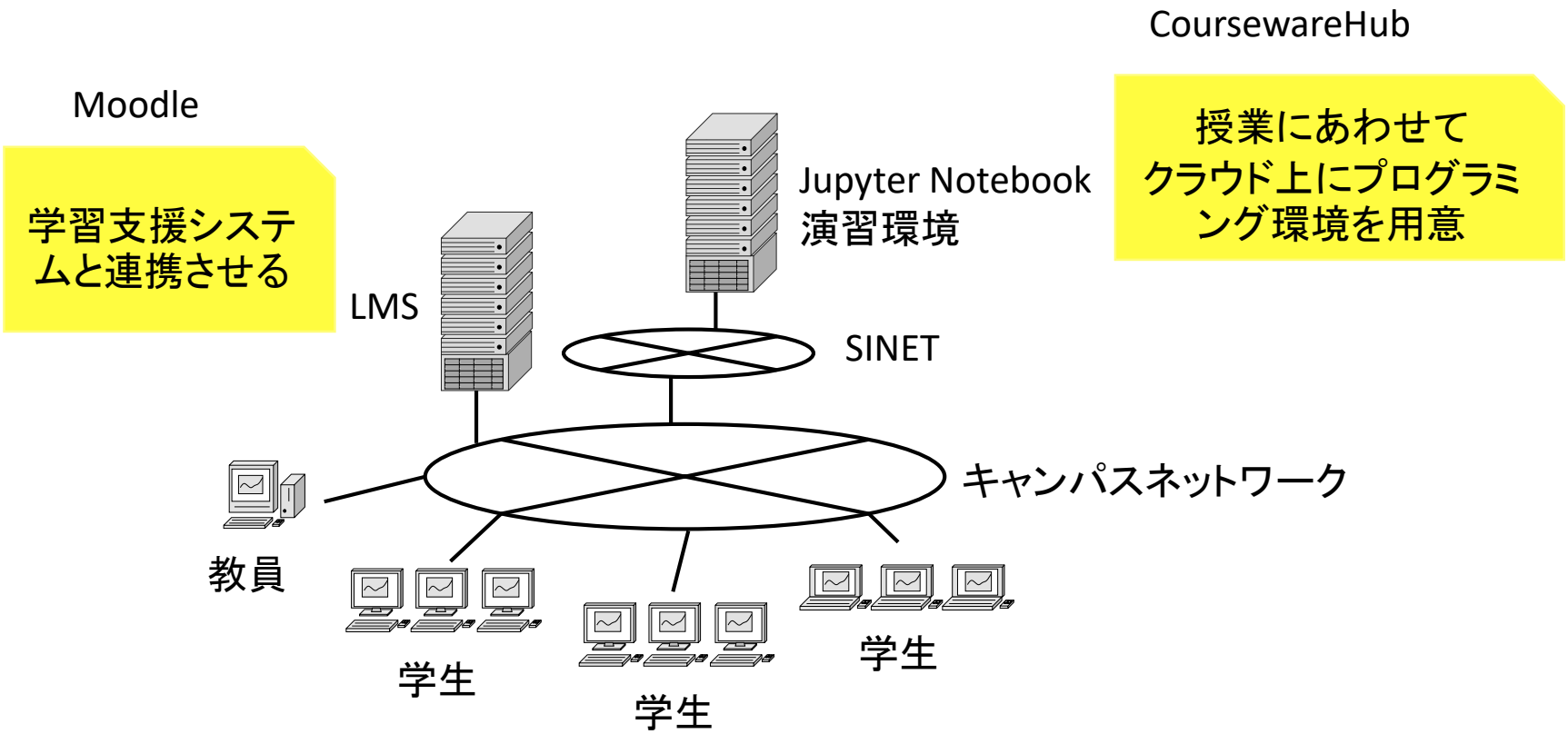
【特徴】

- ブラウザから計算を行う
- Python, R, Julia, Scalaなど複数の言語処理系がサポートされている
- プログラムを作らなくても，大規模計算や機械学習などのライブラリを利用して計算をえる.
- 結果の可視化
  - 説明をマークアップ言語で記述
  - グラフや画像イメージ，ビデオなどをWebページ上にインラインで挿入する
  - 計算結果を直感的に理解しやすい形式で記録することが可能である.
- 作成したコンテンツ(Notebook)は，さまざまな形式で公開し，第三者と共有可能である.

Jupyter Notebookの画面構成

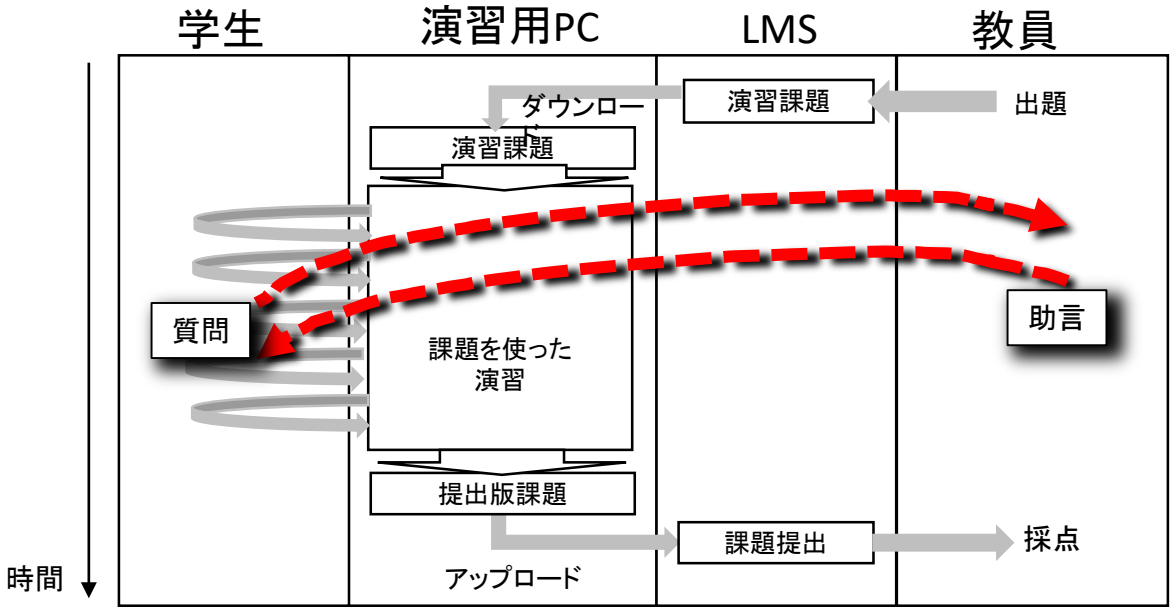


クラウドを利用したプログラミング環境



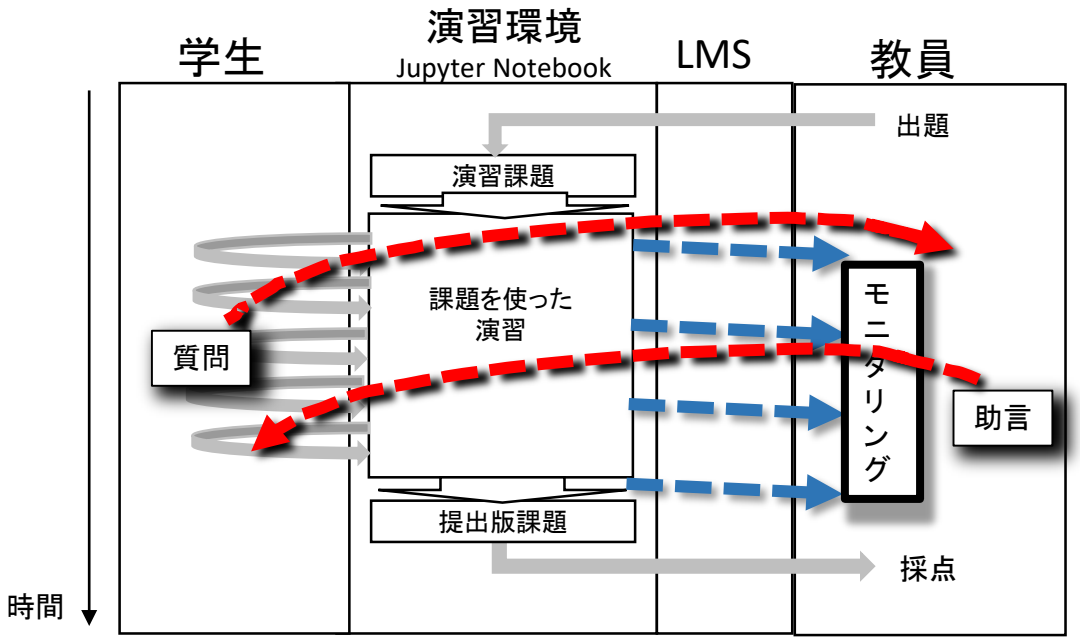
従来の演習環境

- 学生の演習用PCの状況（進捗状況やエラーなど）は分からなかった。



提案する演習環境

- 演習環境を一元管理することで、状況のモニタリングが可能になる。



Jupyter Notebookを使った教材の構成

1. 説明
   - 演習内容やその前提知識の説明を，マークダウンセルに記述する.
   - 学生の理解補助のため，数式や図を併用する.
2. 記述済みのコードセル
   - 例題として，記述済みのコードを含むセルを用意する.
   - 学生はコードをその場で実行して，結果を確かめることができる．また，学生が内容を変更して再実行することで異なる結果が得られることを確認できる.
   - 更に，故意にエラーの出るコードを用意しておき，学生に原因を考えさせる応用も考えられる.
3. 部分記述済みのコードセル
   - コードの一部のみを記述しておき，未記入の部分を学生が記述することでプログラムを完成する.
   - 複数のステップ（セル）に分割してコードを作成する.
4. 未記入のコードセル
   - 自由にコードを書けるように，未記入のセルを用意する.
   - これまでの説明を理解したかどうかを確認するために利用すると効果的である.
   - 単元ごとに設ける提出課題は未記入のコードセルを利用する.

模擬授業

5. 模擬授業の概要

【目標】
- 「プログラミング入門」は新たに全コースの必修科目として設けられた授業であり，１年の後期に開設される.
- プログラミングに必要な概念を理解し，基礎的なプログラムを作成することができるようになることを目標としている.
- コースの開設に先立ち，内容の確認のため模擬授業を実施した.
- 全１５回のうち、模擬授業では7回を実施.

【主な確認項目】
- 教科書や教材の内容確認
- 授業方法に関する評価
- 授業内容のレベル確認

実施内容とスケジュール

回	実施日	模擬授業内容
第1回	6月13日	イントロダクション
第2回	6月20日	プログラミングの基本概念
第3回	6月27日	条件判断
第4回	7月4日	制御構造，繰り返し
第5回	7月6日	データ構造
第6回	7月6日	関数
第7回	7月6日	アルゴリズム1

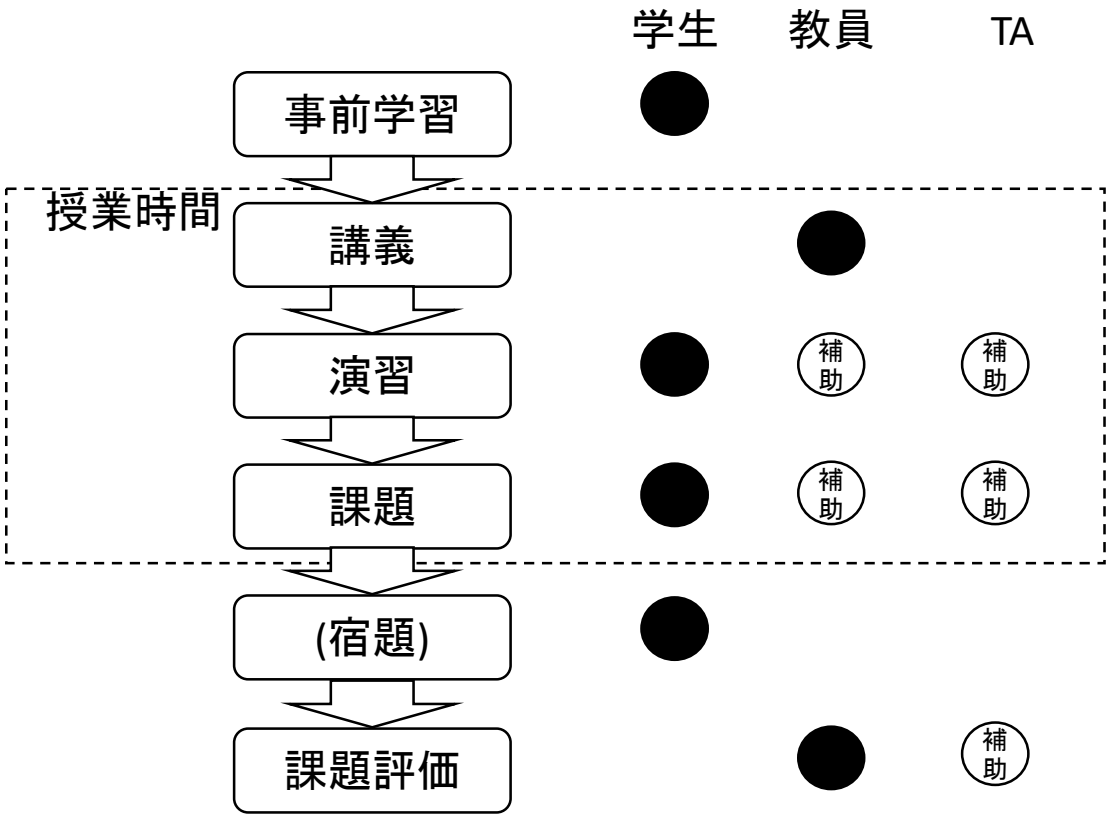
参加者の役割と人数

アクター	人数	実施内容	選定条件
教員	4	講義，演習/課題支援，課題採点	授業の担当教員
TA	2	演習の支援，採点補助	情報系の大学院生
学生	12	授業を受講	非情報系の学部2年生（アルバイト）

授業の進め方

項目	想定時間	内容
事前学習	-	• 学生が，教科書の該当項目を読んで，内容を理解しておく
講義	15分	• 教員が，ポイントとなる項目を説明する.
演習	45分	• 予め用意しておいた演習用Notebookを学生が各自で解くことで理解を深める. • TAおよび教師は学生からの質問を受ける.
課題	30分	• 演習内容をまとめた，課題用Notebookを学生が各自で解いて，提出する. • 教員またはTAが課題用Notebookを評価し，次回以降学生にフィードバックする.
(宿題)	-	• 授業時間内に課題が終わらなかった場合に宿題とする.
課題評価	-	• 教員またはTAが課題を評価してフィードバックを行う.

授業の流れ



模擬授業でのNotebookの構成

回	教科書(文献6)	説明スライド	演習用Notebook	課題用Notebook
第1回	第1章	27ページ	10セル	課題なし
第2回	第2章	17ページ	27セル	1セル
第3回	第3章	19ページ	25セル	1セル
第4回	第4章	14ページ	20セル	3セル
第5回	第5章	16ページ	31セル	5セル
第6回	第6章	10ページ	21セル	5セル
第7回	第8章	17ページ	13セル	5セル

課題の例

jupyter lesson-01 Last Checkpoint: 2019/01/24 (unsaved changes)

Logout Control Panel

File Edit View Insert Cell Kernel Navigate Widgets Help Trusted Python 3 (LC_wrapper)

Run Markdown lang:ALL

Contents

プログラミング入門

第一回　イントロダクション

次のセルに、自分の名前と学籍番号を記入して下さい。

学籍番号：　　氏名：

【進め方の注意】

- 文書を読んで、手順に沿って問いに答えながら進めて下さい。
- 最後の確認問題を解いて、ファイルを提出して下さい。
- 途中わからないことがあれば、教員またはTAに質問して下さい。

1 計算をしてみる

○演習：次の式を評価して計算をしなさい。

（セルの評価を行うには、SHIFT + RETURNを押します）

```
In [1]: 1 + 3
```

* Out [3] Out [2]

```
Out [1]: 4
```

○演習：次のセルで、5 + 4 の計算をしなさい。

```
In [2]: 5 + 4
Out [2]: 9
```

ヒント：結果が 9 になれば正解です。

計算には、整数でなく小数も利用することができます。

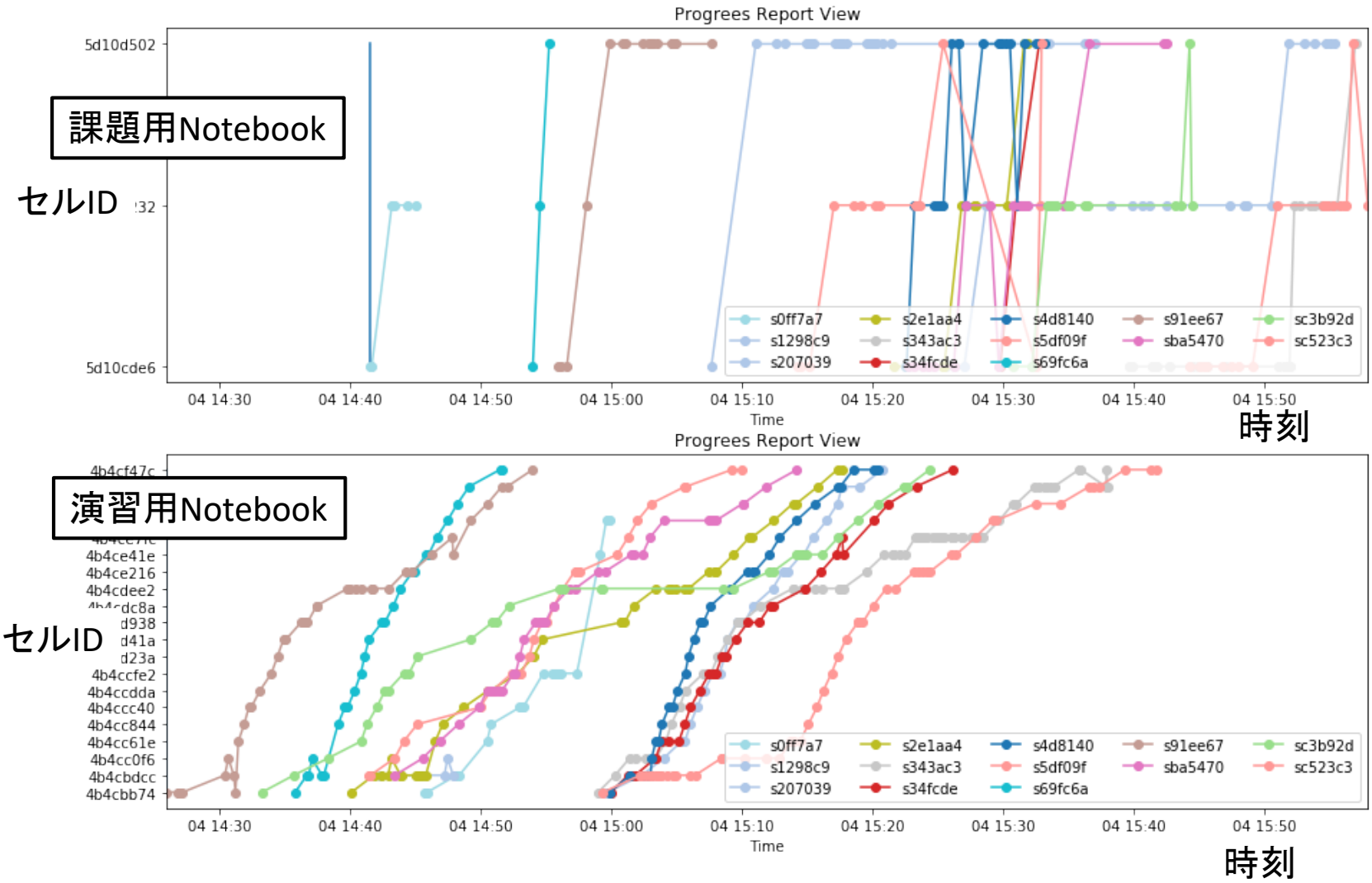
○演習：次のセルで、小数を使って「$3.14 + 4.3$」計算をしなさい。

```
In [3]: 3.14 + 4.3
Out [3]: 7.4399999999999995
```

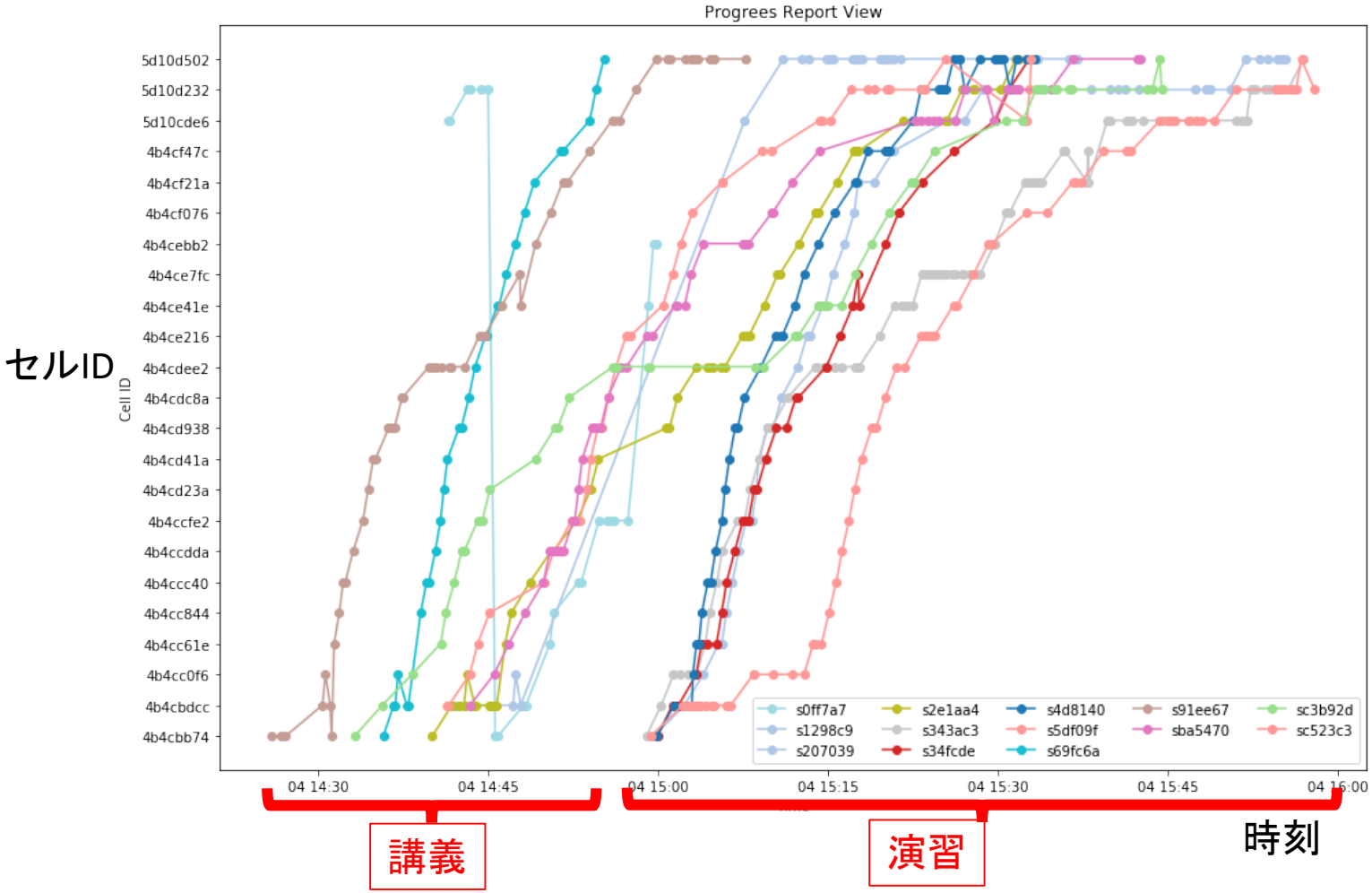
収集したデータ

1. 事前アンケート，事後アンケート(各1回)
   コース開始時と終了時に，Moodleを使いプログラミングに関する知識などの主観的な情報を収集した
2. 事前学習レポート(7回)
   毎回，事前学習の中での不明点などを簡単なレポートとして提出してもらった
3. 模擬授業に関するアンケート(7回)
   Moodleを使い各回の授業に対する主観評価を収集した
4. Notebookの実行履歴(7回分)
   模擬授業7回で，約11,000件のNotebookの実行履歴データをCoursewareHubで収集した
5. 演習Notebookおよび課題Notebook(7回分)
   CoursewareHub の各学生のフォルダに保存されているNotebookを収集した

進捗レポート

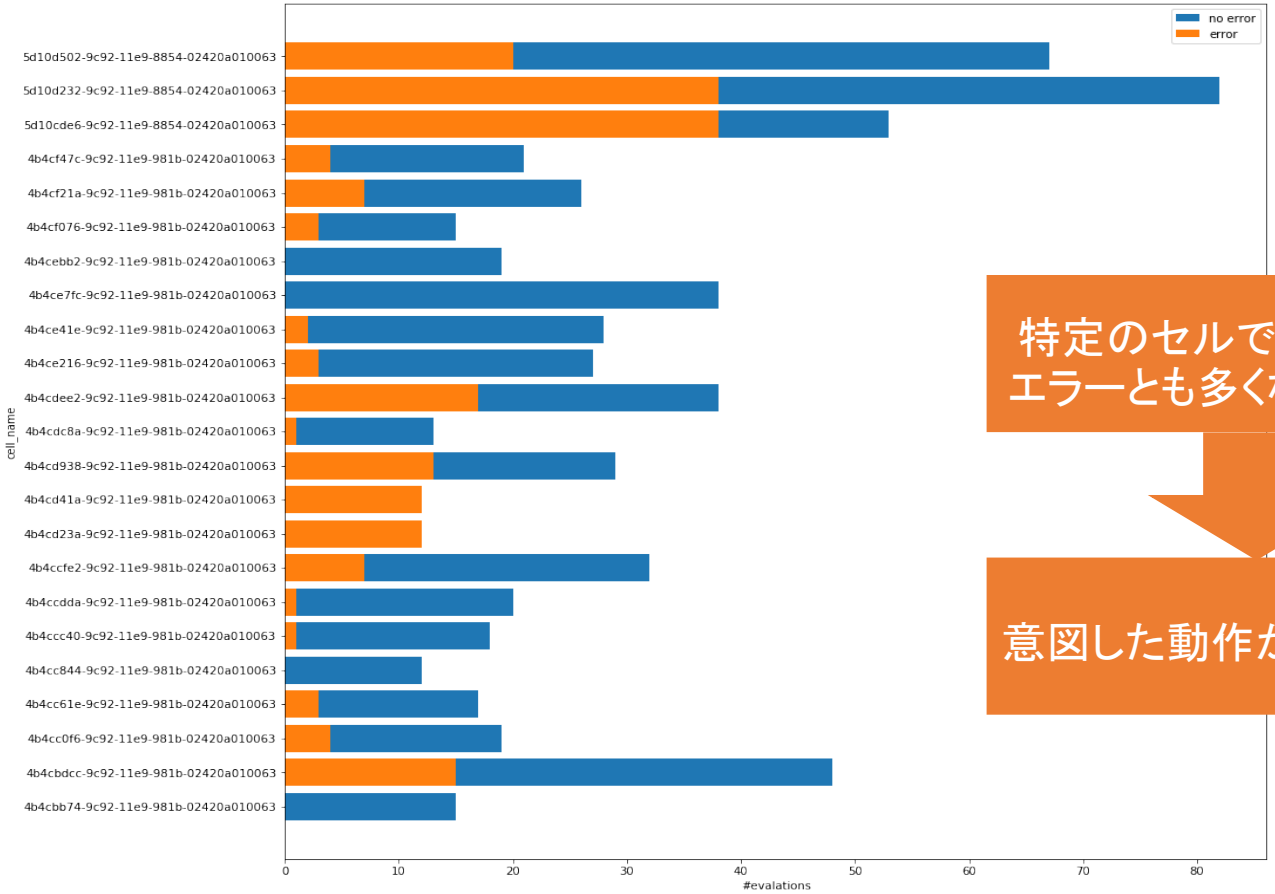


進捗グラフの例（第4回）



セルごとの評価数,エラー数(第4回)

セルID

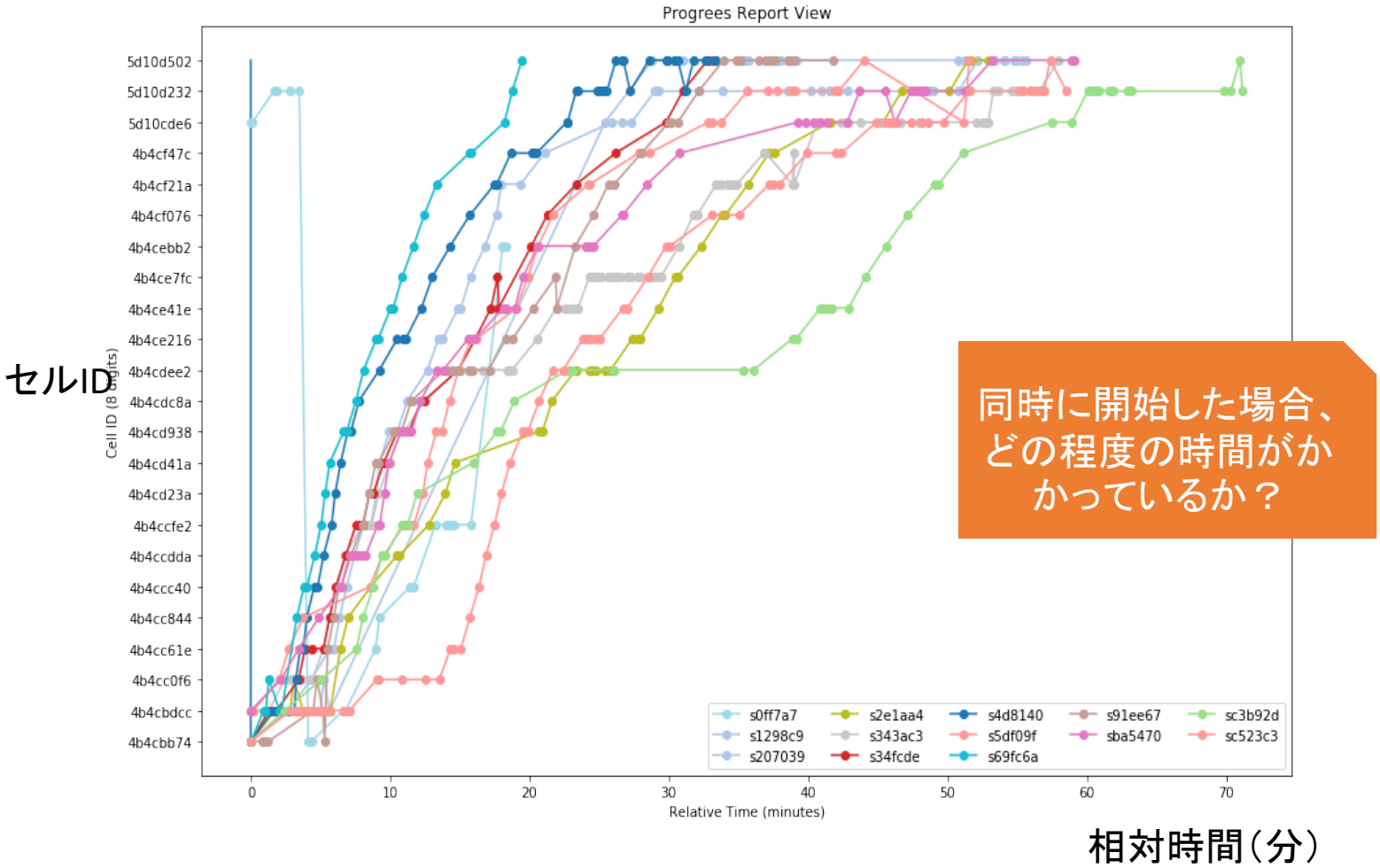


特定のセルで評価回数、エラーとも多くなっている。

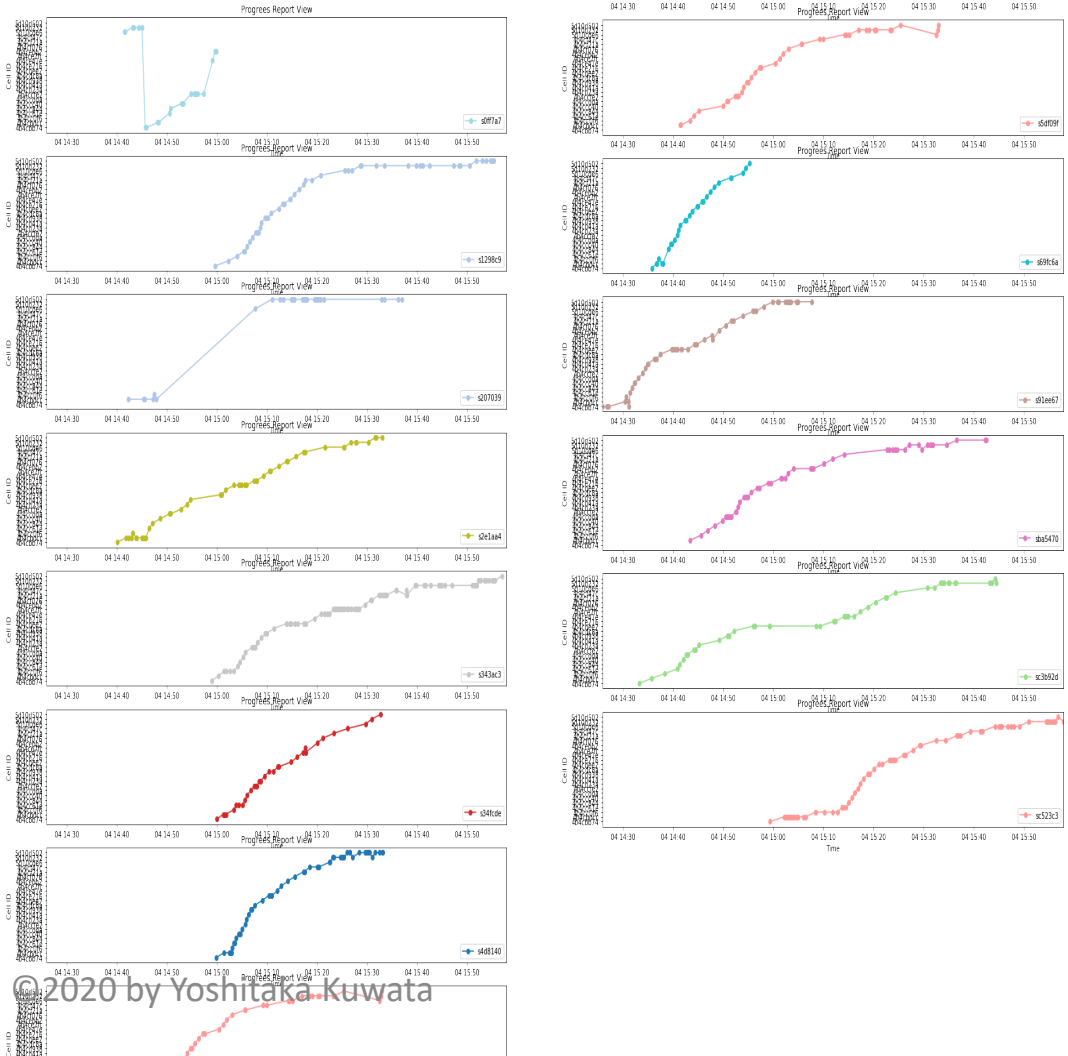
意図した動作か確認する。

評価回数・エラー数

相対進捗グラフ



個々人の進捗状況表



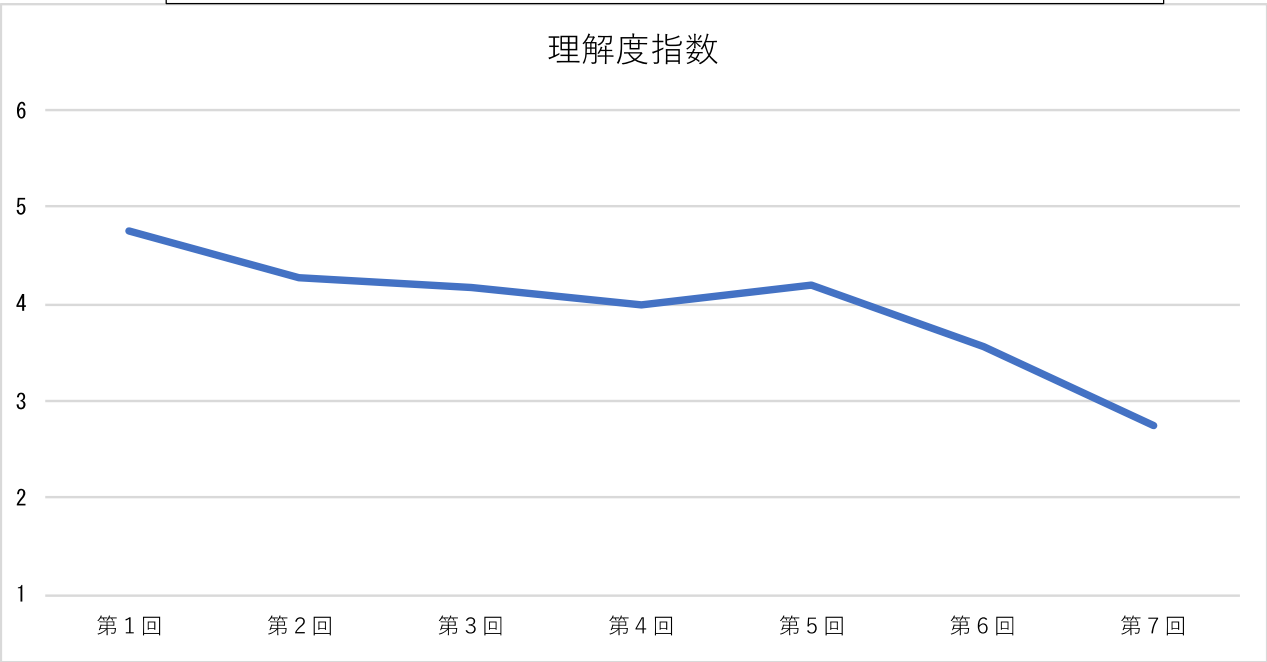
考察：フライングする学生

開始 / 進捗	早い（15:00以前）	普通（15:00以降）
早い（40分以内）	（パタンa）2人	（パタンb）2人
普通（40分以上）	（パタンc）5人	（パタンd）3人

フライングを禁止すべきか？

理解度指数の変化（アンケート調査）

これまでの授業内容は，どの程度理解していますか？
1：全くわからない
2：ほとんど理解していない
3：どちらかというと，理解していない
4：どちらかというと，理解している
5：ほとんど理解している
6：完璧に理解している



2019年度授業

授業の概要

項目	内容
コース名	プログラミング入門
対象者	創造工学科/システム理化学科/夜間主コース　1年生　総計約600名（本研究へのデータ提供の合意者のみ分析を実施）
クラス数	4クラス（昼間），1クラス（夜間）
担当教員数	4人
回数	15回（各回90分）
選択/必修	必修
言語	Python
環境	Jupyter Notebook (JupyterHub)
授業形態	講義/演習（反転学習）
ティーチングアシスタント(TA)	学生約30名に対して1名を配置
教科書	Jupyter　Nortebookで始めるプログラミング2019

授業の実施方保

No.	内容	方法	時間
-	事前学習	教科書	前日まで
1	前回小テストの答え合わせ	スライド説明 (Moodle資料配布)	5分
2	講義	スライド説明 (Moodle資料配布)	10-15分
3	演習	CoursewareHubで配布	20-30分
4	課題	CoursewareHubで配布，回収，採点，返却	30-40分 (宿題)
5	小テスト	Moodleで実施	5分

授業計画 （1）

- 教科書に沿って進める
- 総合演習を設けて復習をする

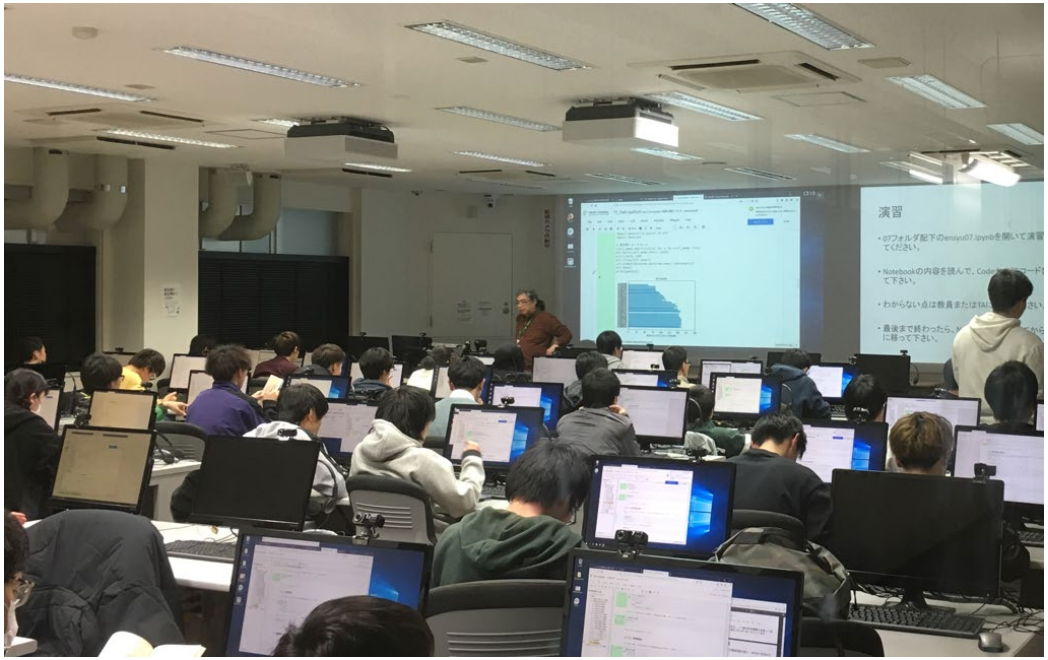
回	表題	内容	教科書
1	イントロダクション	授業の進め方、プログラミング言語の紹介、授業で利用するプログラミング環境の紹介	第1章
2	プログラミングの基本概念	データ型、入出力、演算子、逐次処理	第2章
3	条件判断	条件判断と分岐処理	第3章
4	制御構造、繰り返し	制御構造、繰り返し処理	第4章
5	総合演習1	繰り返しや条件判断を題材とした総合演習課題	第1−4章
6	データ構造	基本的なデータ構造（リスト、辞書）	第5章
7	関数	関数の概念、ローカル変数、再帰呼び出し	第6章
8	総合演習2	関数やデータ構造を題材とした総合演習課題	第5−6章

授業計画 （2）

回	表題	内容	教科書
9	可視化	ライブラリの利用、グラフの作成、統計処理	第7章
10	アルゴリズム（1）	ソート	第8章
11	アルゴリズム（2）	線形探索と二分検索	第9章
12	総合演習3	アリゴリズムの総合演習	第7―9章
13	シミュレーション（1）	酔歩問題	第10章
14	シミュレーション（2）	モンテカルロ法	第11章
15	総合演習4	シミュレーションを題材とした総合演習課題	第10―11章

演習の様子

- 大学実習室のPCを利用
- 複数教室で演習を同時に実施
- 教員、TAが学生からの質問に対応



学生への説明（授業教材）

利用するプログラミング環境 (Jupyter Notebook)

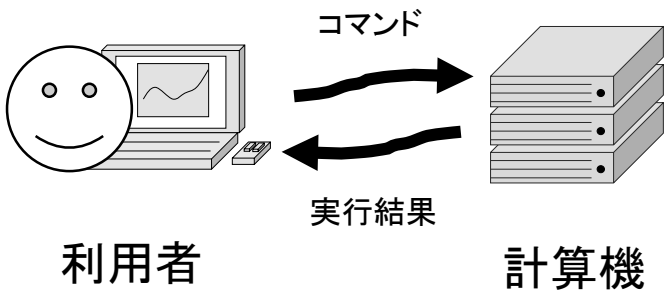
Webブラウザを利用した対話的コンピューティング環境。

- Jupyter Notebookでは、Pythonで書かれたプログラムの呼び出しや、データの操作により対話的コンピューティングを行うことができる。

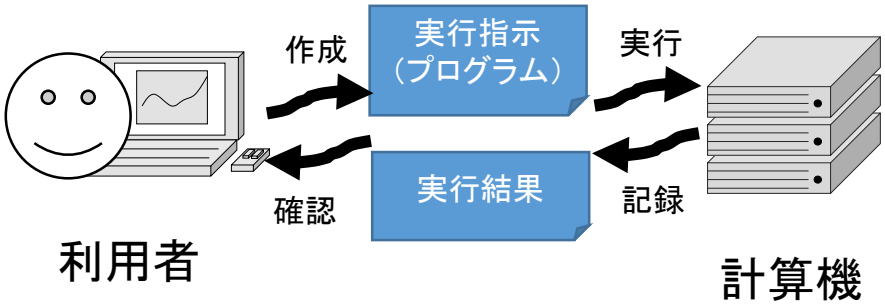
対話的コンピューティング

- 対話的コンピューティング
  利用者が逐次的にコンピュータにコマンドを入力し、その出力結果を確認しながら、次のコマンドを実行してゆく方法
- バッチ処理的なコンピューティング
  予めプログラミングを作成しておき、コンピュータに実行を指示する方法

対話的コンピューティング



バッチ処理的コンピューティング



学生への説明（授業教材）

Jupyter Notebook環境についての説明

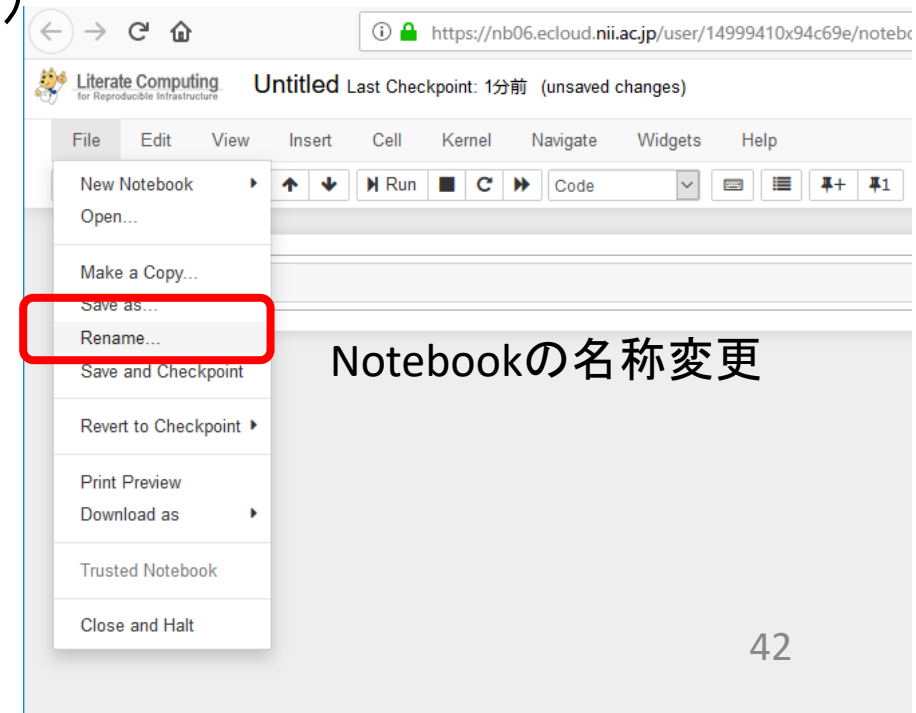
- コースウエアハブ
  - 国立情報学研究所の提供するJupyter Notebook環境
  - 教育・研究用に提供されている。
  - クラウド上のサービスとして提供されるため、Webブラウザから利用可能。

- 注意点
  - 学内からのみアクセス可能
    - 実習室および図書館のパソコン
    - 無線LAN経由の接続

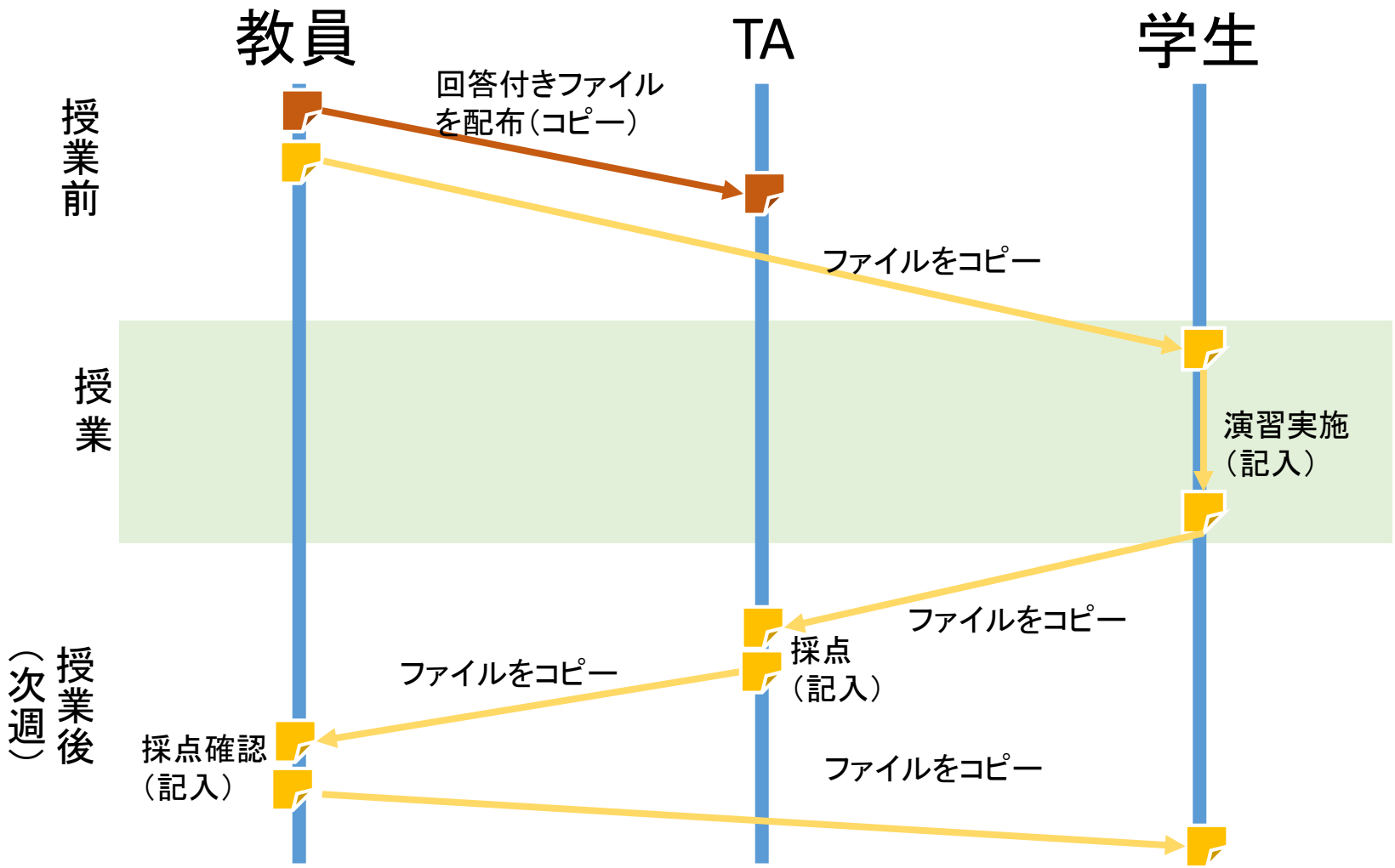
学生への説明（授業教材）

課題の提出

- Notebookの新規作成
- Notebookの名称変更
  Untitled→ kadai01
- 学籍番号、氏名の作成（Markdownセル）
- 表題の作成（Markdownセル）
  - 先頭に移動する
- Codeセルの作成
- 式の記述
  - 一年は何時間か計算する。
  - 一年は何週間か計算する。
- 保存



教材処理の流れ



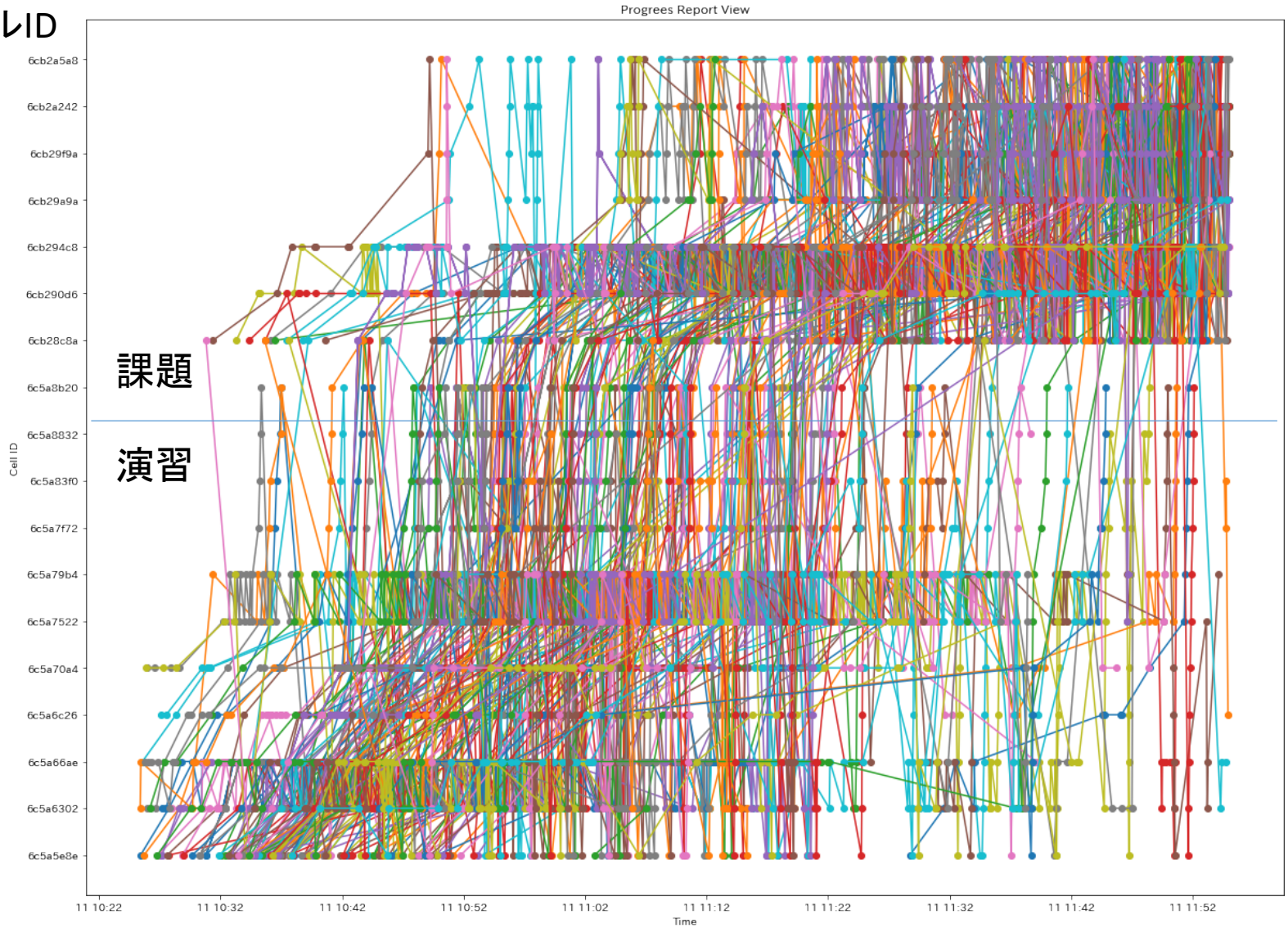
授業の分析

定量データ(Cクラス)

					Cクラス			
講義番号	表題	コードセル数			参加学生数	評価数（授業時間内）		
		演習	課題	合計	（当日）	演習	課題	総評価数
1	イントロダクション	0	2	2	152	0	393	393
2	プログラミングの基本概念	27	1	28	159	8658	1597	10255
3	条件判断	25	1	26	164	7501	576	8077
4	制御構造、配列	20	3	23	153	6328	2000	8328
5	総合演習１	28	4	32	156	10483	767	11250
6	データ構造	17	4	21	158	4041	1651	5692
7	関数、再帰呼び出し	45	6	51	158	12874	4028	16903
8	総合演習２	23	4	27	158	10486	1107	11593
9	可視化	22	5	27	159	4919	4581	9500
10	アルゴリズム１	11	7	18	164	5374	6989	12363
11	アルゴリズム２	19	7	26	155	7988	4326	12314
12	総合演習３	16	7	23	154	7855	4113	11968
13	シミュレーション１	12	4	16	156	3134	1924	5058
14	シミュレーション２	17	10	27	158	3683	5026	8709
15	総合演習４	0	14	14	150	0	13484	13484
	合計	282	79	361	2204	93324	39078	132403

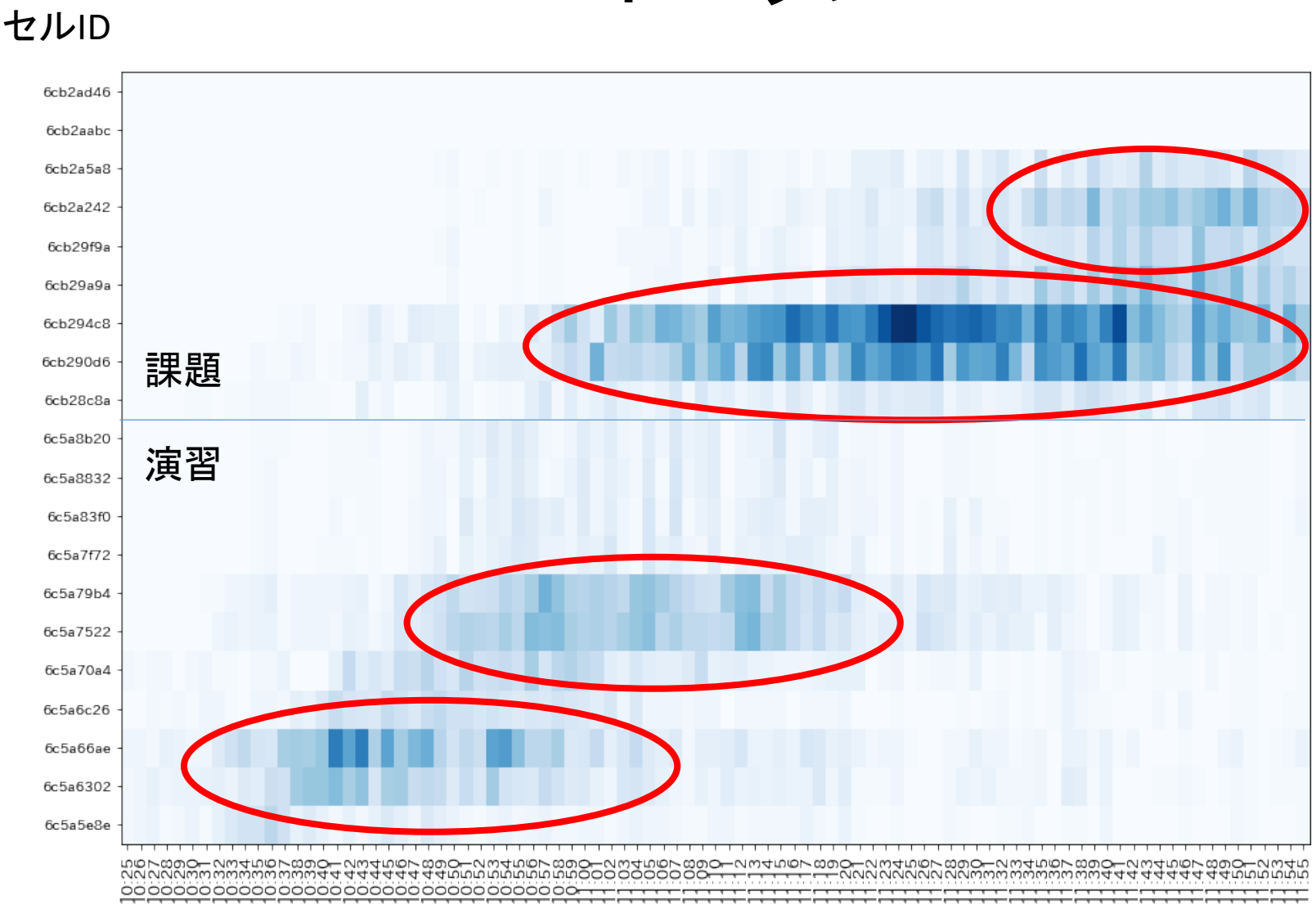
実験参加に合意していない学生も数に含む　※15回は合意者のみ

セルID



時刻

ヒートマップ



演習中のエラーの分析

- Python処理系の出すエラーに注目
- 全体のセル評価の3割程度がエラーになっている
- エラーに注目した理由
  - 内容が明確であるので解析しやすい
  - 予め共通の傾向がわかると対応しやすい
  - 全てのエラーの記録がある
- デメリット
  - 全体の3割しかカバーしていない
  - エラー解析でどこまでわかるのか不明
  - エラーメッセージの詳細な体系化がされていない

エラーの例

```
In [5]: fsdsdf 8sd sdf

  File "<ipython-input-5-73ebae2a5291>", line 1
    fsdsdf 8sd sdf
           ^
SyntaxError: invalid syntax
```

traceback message

error

```
In [7]: df = sdfsdf

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-a33d01ab307f> in <module>()
----> 1 df = sdfsdf

NameError: name 'sdfsdf' is not defined
```

traceback message

error

エラーログの解析

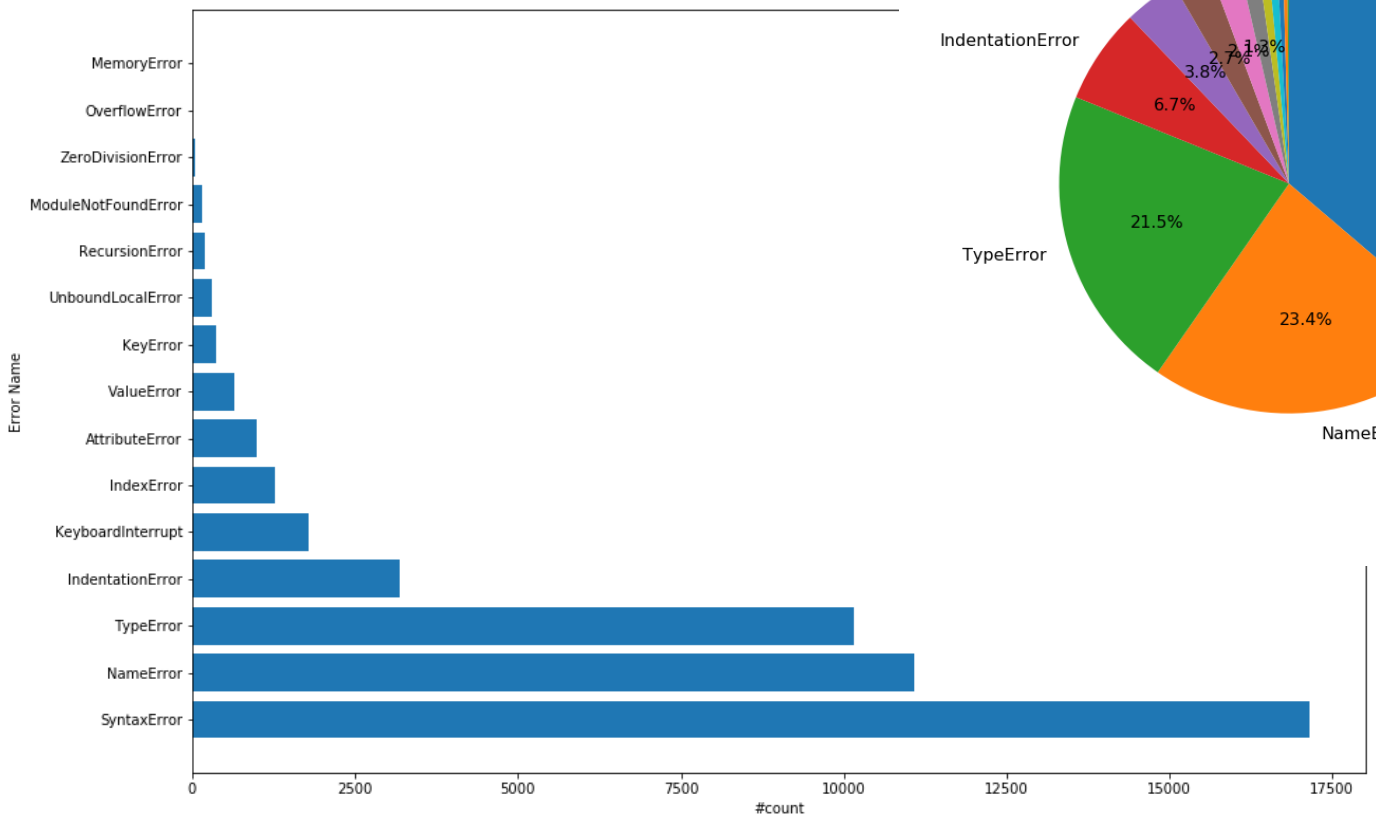
1. エラーのカテゴリ分類
   - Pythonのエラーカテゴリ別

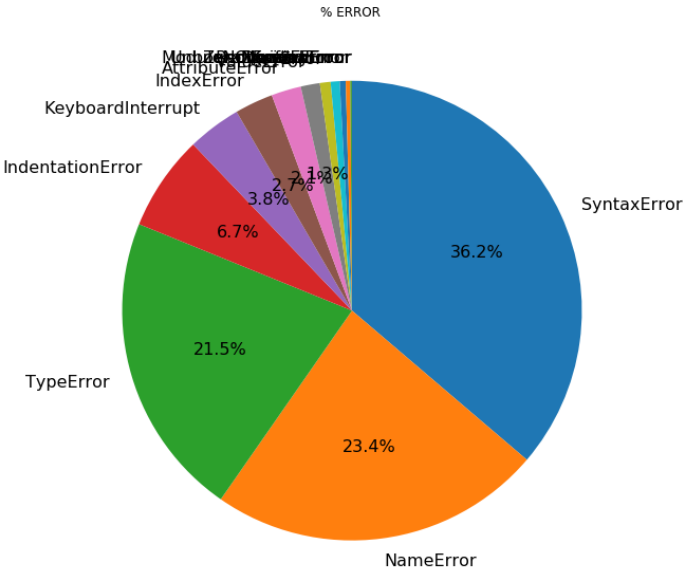
2. tracebackの解析
   - アイキャンデー
     - エラー個所を示すためのキャラクタ（スペース＋”^”など）
     - カラフルな表示のため、エスケープシーケンスが含まれる
   - 自然言語で記載されているため、分類が必要
   - 型や変数名がトレースメッセージに埋め込まれている

→トレースメッセージの統合化（集約化）および類型化が必要

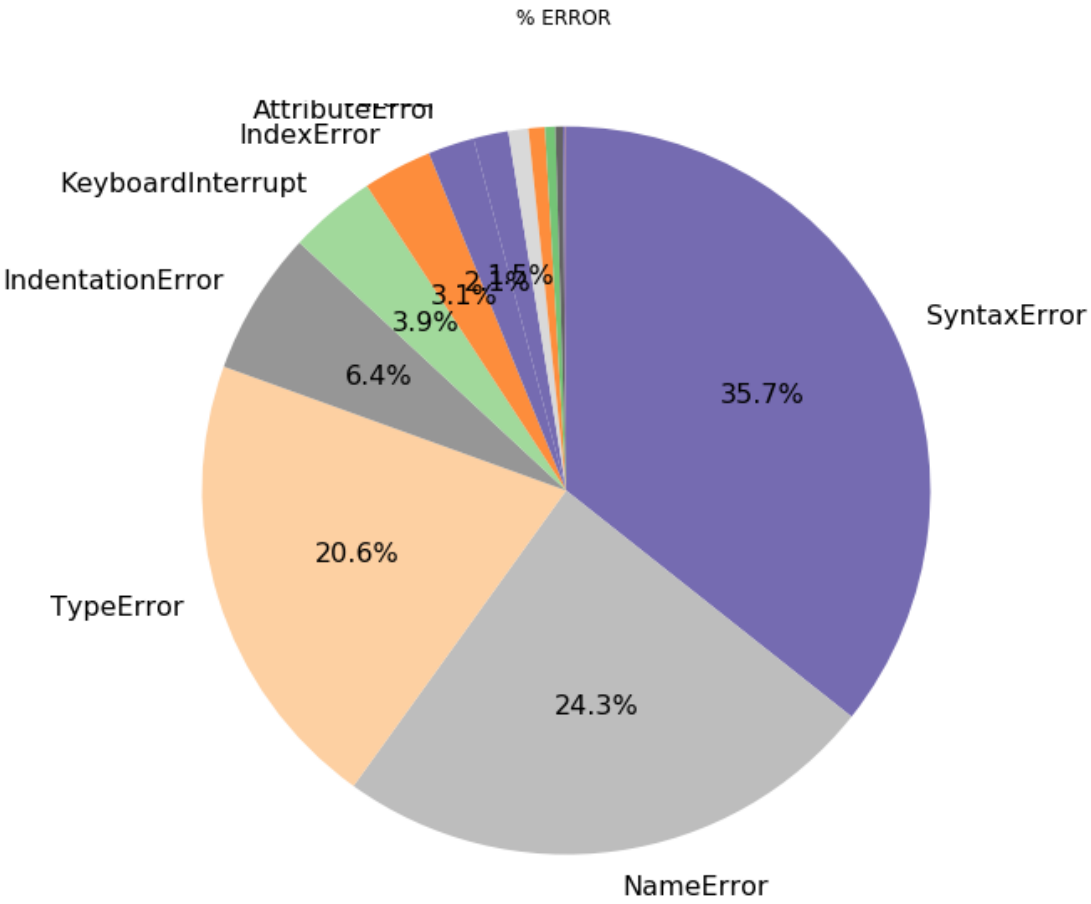
カテゴリ別エラー分類

全15種類が発生（Cクラス）

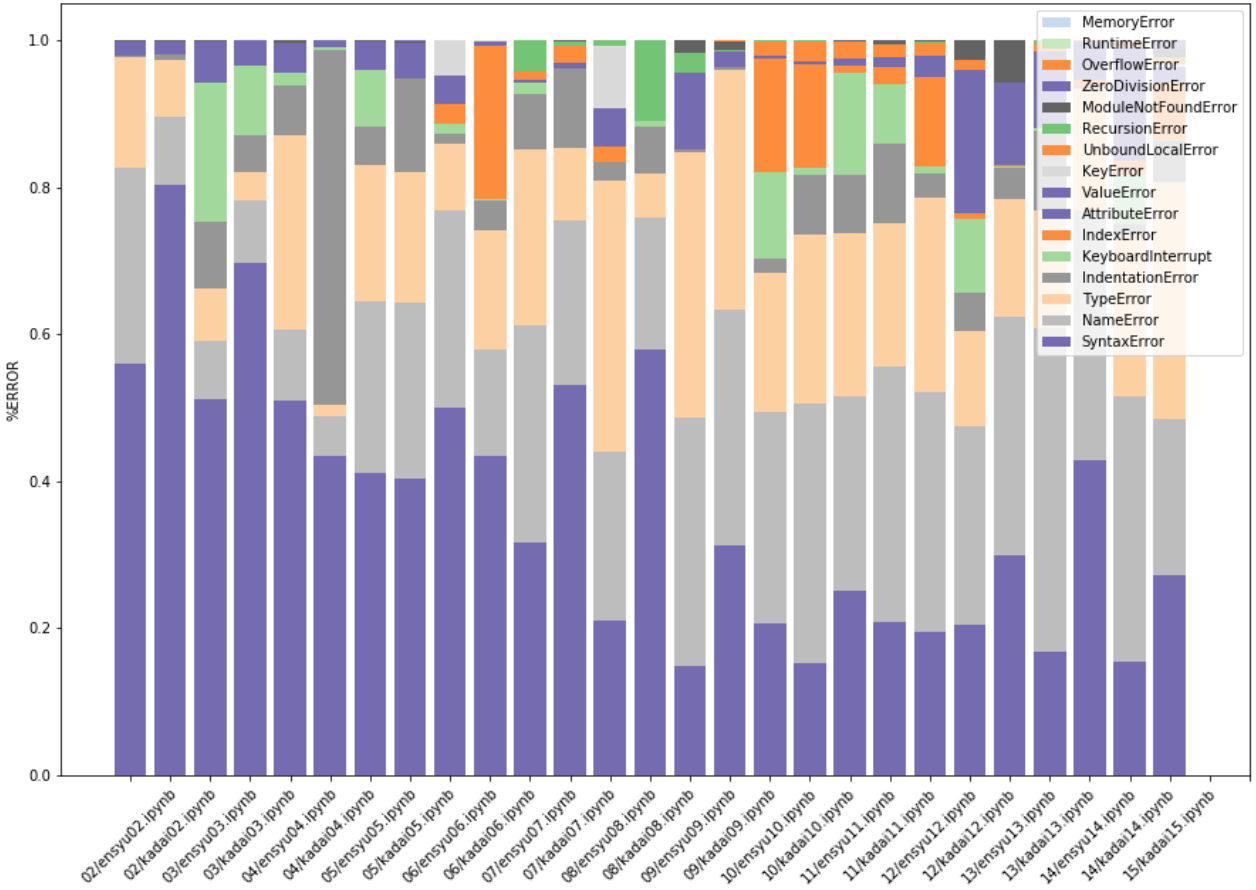




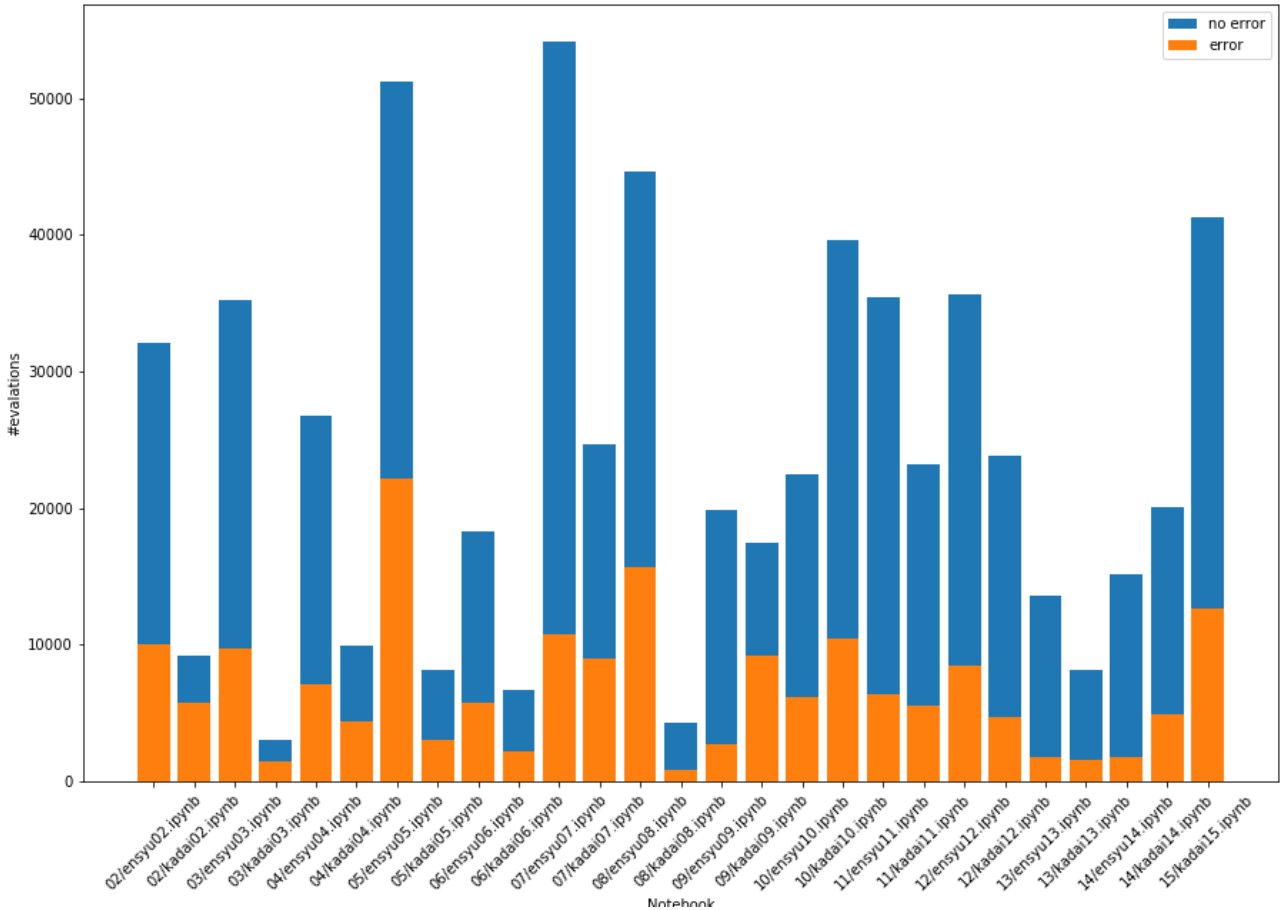
エラー種類と割合



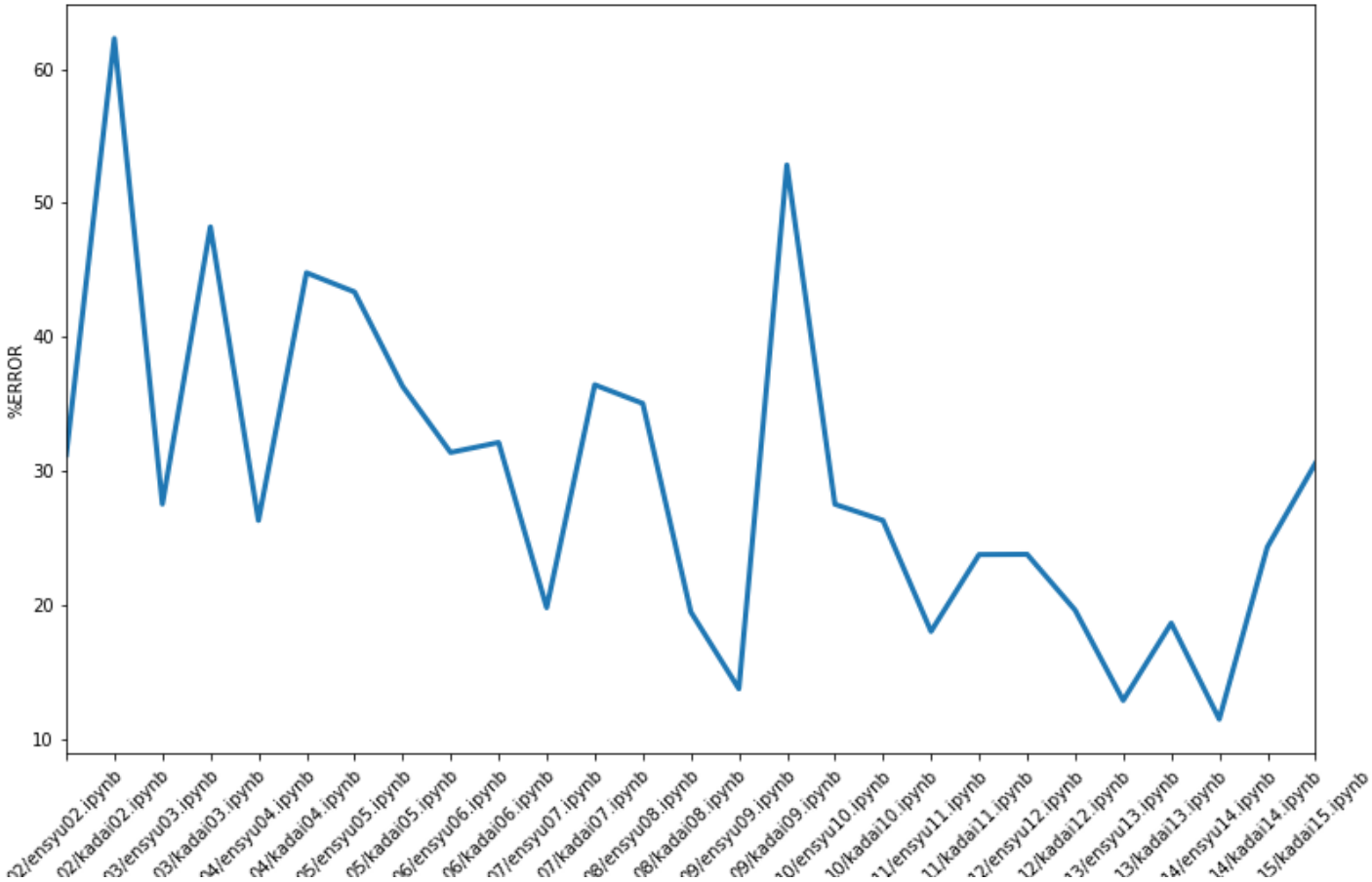
教材ごとのエラーの種類



教材ごとのエラーの割合



エラー率の時系列変化



まとめと今後の課題

まとめ

1. NIIのCoursewareHubを使って大規模なプログラミング授業を実施した。
   - ６００人規模の授業に対応
   - 大きなトラブルなく運用できた
   - 演習の結果を定量的なデータとして収集し、分析した

2. データを活用して演習の支援をした。
   - リアルタイムでの教員支援
   - 次回以降のフィードバック
   - 教材への反映

今後の課題

1. 収集したデータのさらなる活用
   - リアルタイムでの学生支援
   - エラー分析結果の反映

2. ポストコロナ授業
   - ハイブリッド授業
   - 学外からのアクセスの仕組みの確立

本研究はJSPS科研費 (JP18K11561)の「クラウドを活用したプログラミング演習環境に関する研究」の助成を受けたものである