

# 情報リテラシ第二 (1b)

## 第5週

プレゼンテーション (批評)

グラフ作成 (実習C)

文書作成 (段落と数式)

# 課題: プレゼンテーション (スライド批評) 指定されたスライドに対して改善案を書け

## 手順

1. 講義webページにある「第1次スライド」と「批評割当表」をダウンロード
  - 割当表「提出状況」のYES/NOは本人のスライドの提出状況
2. 自分の学籍番号の行の割当「あ」「い」「う」「え」が割り当てられた4つのスライド番号である。該当するスライドに対して改善案を書け
  - アニメーションを使っている場合があるので、「スライドショー」で見るとよい
  - 次回に「この改善案は有益でしたか?」という逆評価を行う
  - 良い点も「良い」と指摘せよ
3. 改善案は[オンラインのページ](#)に入力
  - 期限: 7月13日(金)

# (例) 加計学園疑獄の概要

- ▶ 安倍晋三総理の知人が理事長を務める学校法人「加計学園」が、国の国家戦略特区に獣医学部の新設を申請し、2016年11月に認められ、開設は2018年4月の予定。
- ▶ 獣医学部の新設予定地は、広さ16.8ヘクタール、評価額36億7500万円もの土地なのに、今治市からタダで同学園に譲渡されている。同学園には愛媛県と今治市が96億円もの補助金を支払っている。
- ▶ 国家戦略特区を担当する内閣府と、実際に学部の新設を認可する文部科学省が協議している。報道によれば、審査のスケジュールをめぐる難色を示した文科省に対し、内閣府が「官邸の最高レベルが言っている」「総理ご意向」などと開設を急がせた。

# 文章の量

- 注意点

- 聴衆はスライドに書いた文章を読んでしまう
  - 話を聞いてもらえない
- 小さい文字は遠くからは読めない

- とるべき方針

- 重要なキーワードのみにする (1項目はスライド1行程度)
- 口頭で説明することだけ書く
- 最低サイズ: 24ポイント

# (例) 加計学園疑獄の概要

- ▶ 学校法人「加計学園」の申請

- ▶ 申請内容: 獣医学部の新設

- ▶ 設置先: 国家戦略特区

- ▶ 学園への補助

- ▶ 今治市から土地を譲渡 (広さ16.8ha、評価額36億円)

- ▶ 愛媛県と今治市から補助金 (96億円)

- ▶ 政府の対応

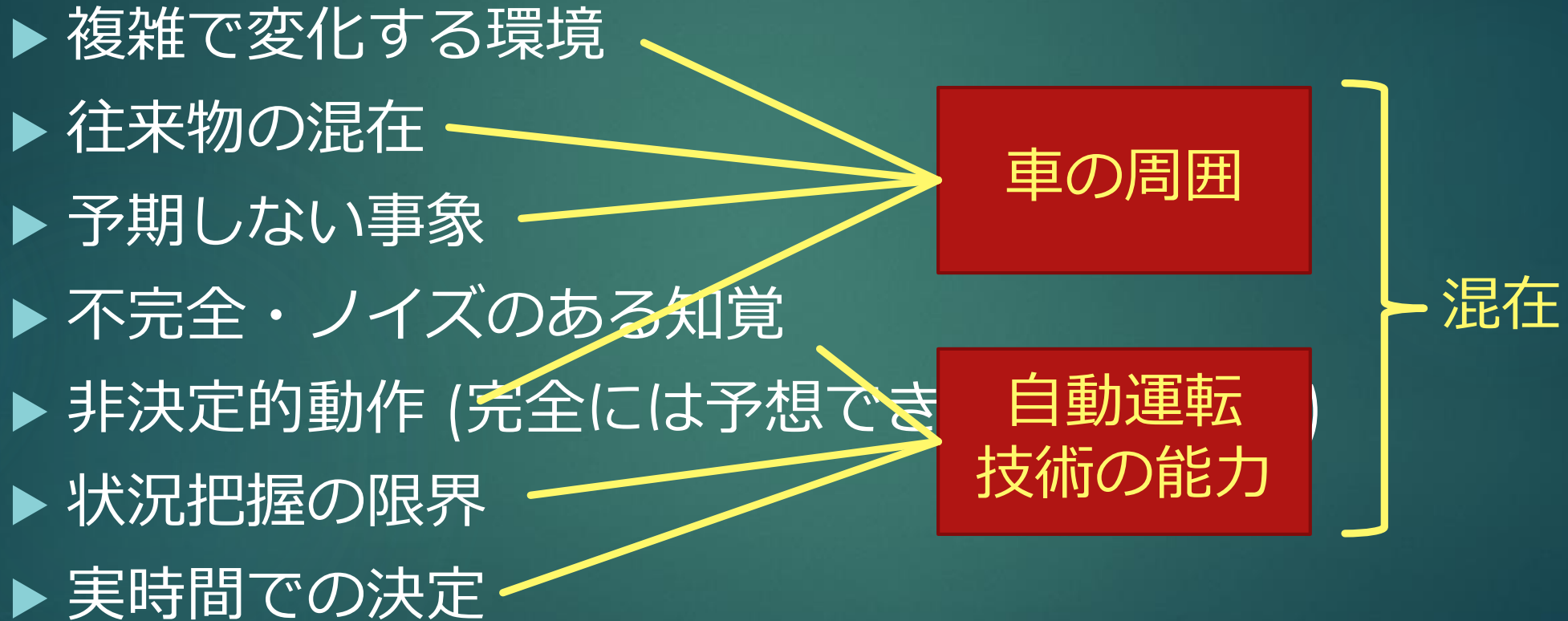
- ▶ 内閣府: 国家戦略特区を担当

- ▶ 文部科学省: 学部新設認可を担当

スケジュールをめぐる対立  
(報道による)



# (例) 自動運転: 安全性の向上?



# 箇条書き

- 原則
  - 並列・直列する事柄を並べる
- 注意点
  - 「同じレベルで並列・直列しているか？」
- とるべき方針
  - 並んでいるものの関係を常に考える

# (例) 自動運転: 安全性の向上?

- ▶ 複雑で変化する環境
  - ▶ 往来物の混在
  - ▶ 予期しない事象
  - ▶ 非決定的動作 (完全には予想できない動作結果)
- ▶ 自動運転技術の不足
  - ▶ 不完全・ノイズのある知覚
  - ▶ 状況把握の限界
  - ▶ 実時間での決定



# *motivating example:* **SPARK**

```
class MyCoolRddApp {  
  val param = ...  
  val log = new Log(...)  
  ...  
  def work(rdd: RDD[Int]) {  
    rdd.map(x => x + param)  
        .reduce(...)  
  }  
}
```

## **PROBLEM:**

(x => x + param)  
not serializable  
because it captures  
this of type  
MyCoolRddApp  
which is itself not  
serializable

# motivating example: SPARK

```
class MyCoolRddApp {  
  val param = ...  
  val log = new Log(...)  
  ...  
  def work(rdd: RDD[Int]) {  
    rdd.map(x => x + param)  
      .reduce(...)  
  }  
}
```

## PROBLEM:

(x => x + param)

*not serializable*

because it captures  
this of type

MyCoolRddApp

which is itself not  
serializable

## motivating example: SPARK

```
class MyCoolRDDApp {  
  val param = ...  
  val log = new Log(...)  
  
  def work(rdd: RDD[Int]) {  
    rdd.map(x => x + param)  
    .reduce(...)  
  }  
}
```

### PROBLEM:

(x => x + param)  
not serializable  
because it captures  
this of type  
MyCoolRDDApp  
which is itself not  
serializable

<http://docs.scala-lang.org/sips/pending/spores.html>

# 色使い

- 注意点

- 多すぎになりやすい
- 作成環境と発表環境の違い (機材・知覚)

- とるべき方針

- 重要なキーワード数個に色を付ける
- 色の種類は必要最小限
- 色が無くても分かるように
- コントラストに注意する


# 色の使い方: コントラストに気を付けよ

- 良い色使い

- 明るい背景 + 暗い文字
- 暗い背景 + 明る文字



プレゼン  
テーション



プレゼン  
テーション



プレゼン  
テーション

- 悪い色使い

- 明るい背景 + 明るい文字
- 暗い背景 + 暗い文字
- (明暗と暗明の混在)



プレゼン  
テーション



プレゼン  
テーション



プレゼン  
テーション

# motivating example: SFARK

```
class MySerializable {
  val param = ...
  def log = ...
}

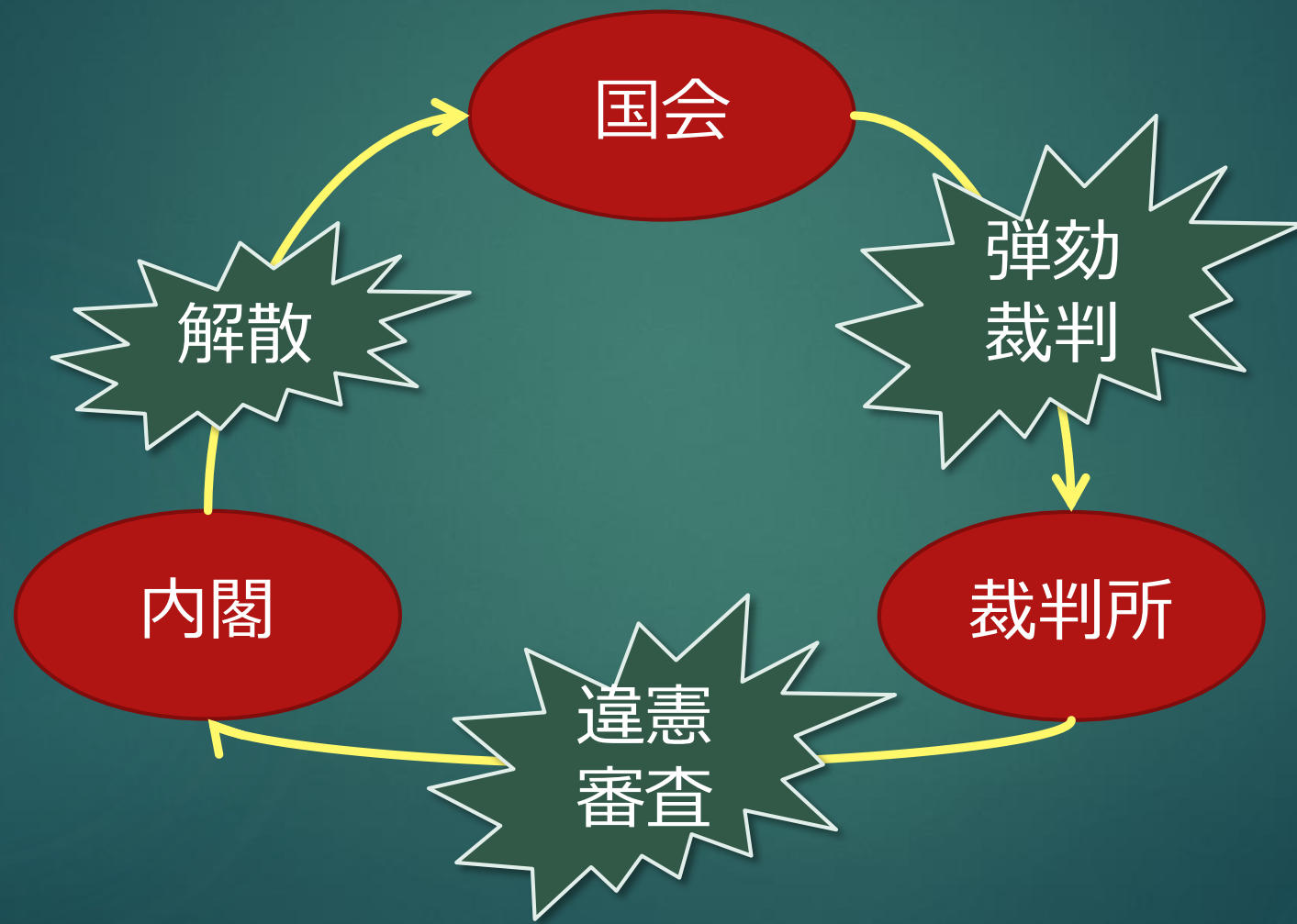
val work = new MySerializable {
  val param = ...
  def log = ...
}
```

**PROBLEM:**  
( $\lambda$   $\Rightarrow$   $\lambda$  = param)  
not serializable  
because it captures  
this of type  
MySerializable  
which is itself not  
serializable

# (例) 三権分立

- ▶ 国会
  - ▶ 弾劾裁判を行う
  - ▶ 弾劾裁判によって裁判官は罷免されうる
- ▶ 裁判所
  - ▶ 法律の違憲審査をする
  - ▶ 審査される法律は内閣が制定
- ▶ 内閣
  - ▶ 国会を解散できる

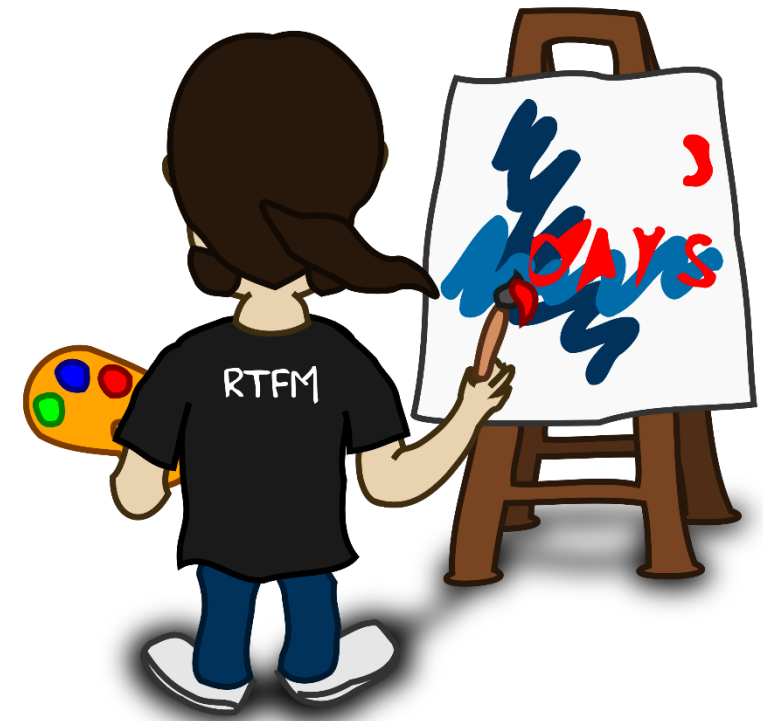
# (例) 三権分立





# 図の使い方

- 原則
  - 関係を示すのに有効
  - 数量を示すのに有効 (グラフ vs 表)
- 注意点
  - 図形の意味は伝わりにくい
  - 一貫性がないと聴衆は混乱する
  - 理解できる要素数には限りがある
  - 挿絵は注意を疎外しないように

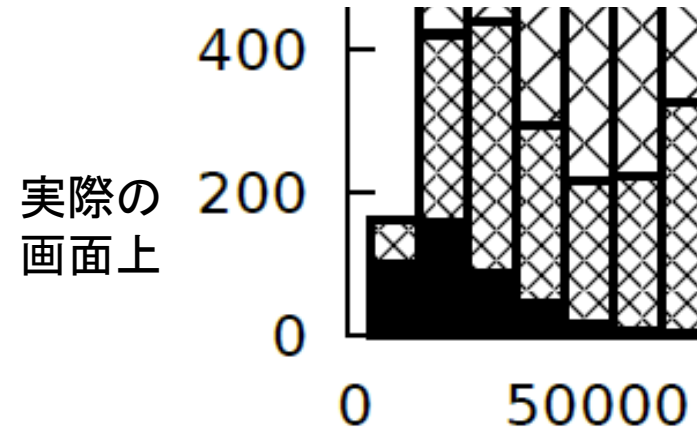
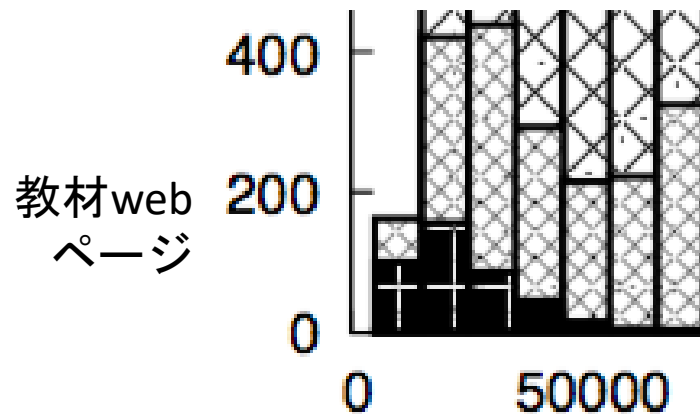


# アニメーション

- 注意点
  - 使い過ぎになりやすい
  - 見逃される
- とるべき方針
  - 順に見せる・隠す
    - 聴衆の注意を集中させたい場合
    - 図の位置関係を変えたくない場合
  - 動きをつける: 付加情報に限る

# グラフ作成: 実習C

1. 図2を作成するための描画レシピを見つける
2. 中を見て、データファイルのパス名と、出力する画像ファイルのパス名を確認
3. gnuplotと描画レシピを使って画像ファイルを作成し、結果を確認
4. 描画レシピを編集して共通教材webページのような出力を得る
  - 注: 黒パターンに見える白い十字線は不要



# 実習Cのヒント

## ヒント

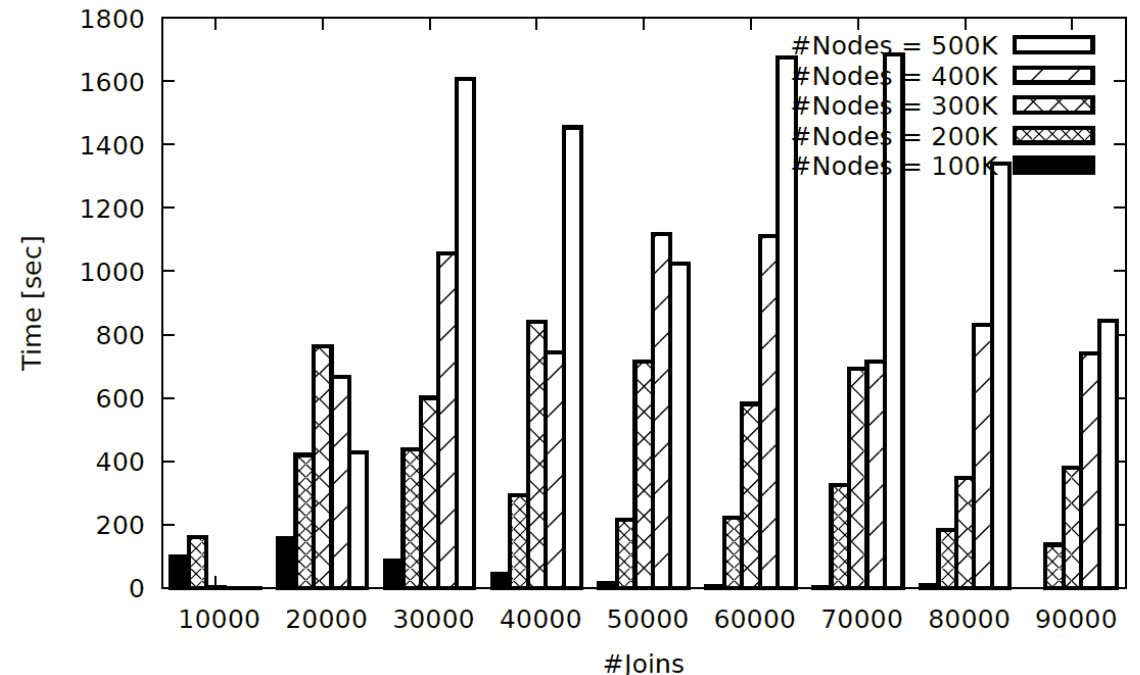
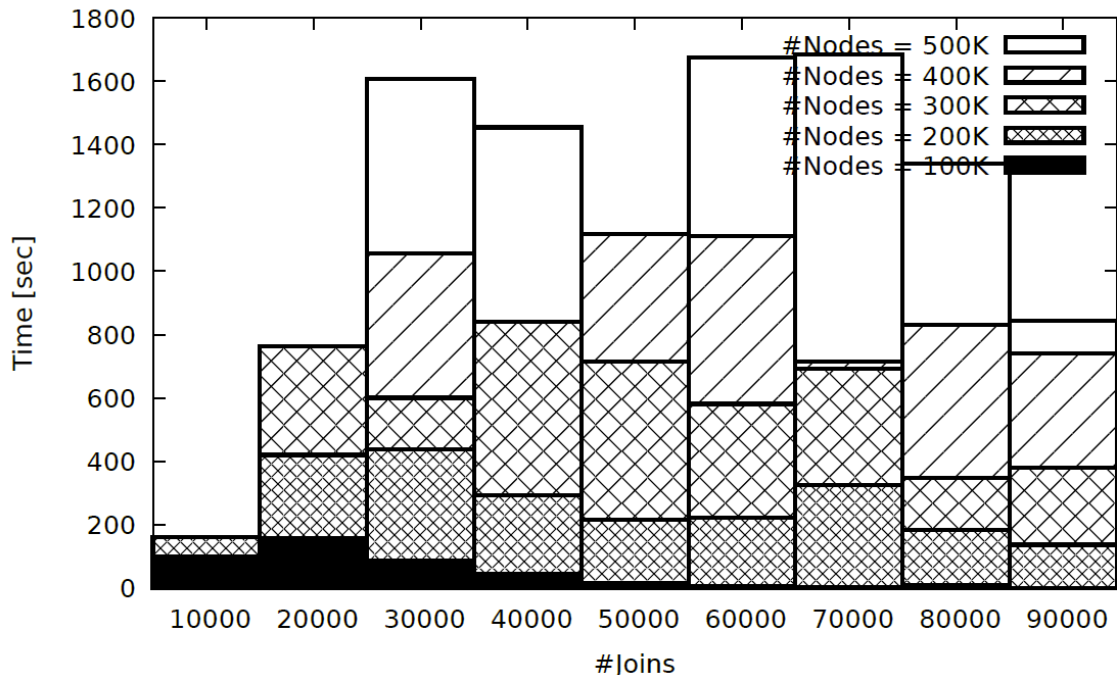
- with boxes: グラフの種類 (棒グラフ、折線グラフなど)
  - [gnuplot入門/スタイル一覧](#)
- linewidth 2: 線の太さ
- linecolor rgb "black": 線や塗りの色
- fillstyle pattern 4: 塗り模様
  - [Gnuplot の描画パラメター一覧](#)

注: 行末のバックスラッシュ記号は「次の行に続く」の意味: 消してはいけない

```
plot \  
  "../cnm/time-join-500K.data" title "#Nodes = 500K", \  
  "../cnm/time-join-400K.data" title "#Nodes = 400K" with boxes, \  
  "../cnm/time-join-300K.data" title "#Nodes = 300K" with boxes linewidth 2, \  
  "../cnm/time-join-200K.data" title "#Nodes = 200K" with boxes linewidth 2 linecolor rgb "black", \  
  "../cnm/time-join-100K.data" title "#Nodes = 100K" with boxes linewidth 2 linecolor rgb "black" fillstyle pattern 4
```

# 実習Cが終わった人への挑戦

- 図2は複数の棒グラフを前後に重ねたもの (左は図2の拡大)  
→後ろの棒が隠れているかも
- 右のように複数の棒グラフを横並びにしてみよう!
  - 参考: [ぷろっとぎゃらりい](#)・[棒グラフを作る](#)



# LaTeXによる文書作成

## 解説予定

- 段落
  - 数式
  - 図、表、参照
  - 箇条書
  - 文献と引用
- 今週
  - 次週
  - 次々週