

岡山大学天文台スカイモニターを用いた
夜空の明るさの季節変化に関する研究

岡山大学 大学院 自然科学研究科 地球科学専攻
41M22504 福本佑典

2024/02/09

要旨

岡山大学天文台スカイモニターが撮影した画像を用いて、夜空の明るさの季節変化と、緊急事態宣言の夜空の明るさへの影響を調べた。太陽・月の影響がなく、かつ、雲がない画像を選別し、2019年7月から2023年12月の約4年半の期間から288夜について夜空の明るさを測定した。岡山の夜空の明るさには、3月から5月の春に明るく、8月から1月の秋から冬に暗いという明瞭な季節変化があることが明らかになった。また、PM2.5と夜空の明るさに正の相関があることが見つかった。今回の解析では、緊急事態宣言が夜空の明るさに与えた影響は見えなかった。

Abstract

Using sky images taken by the Sky Monitor at Okayama University Observatory, we investigated seasonal variation in the brightness of the night sky and the effect of declaration of a state of emergency on the brightness of the night sky. First, we extracted images that were not affected by sunlight, moonlight, and clouds. Then, we evaluated the brightness of the late-night sky by extracting images taken during the late-night hours. The brightness of the night sky was measured for a total of 288 nights from images taken between July 2019 and December 2023. Our result clearly indicates that the night sky in Okayama is bright in spring (March to May) and dark in autumn and winter (August to January). We also found a positive correlation between PM2.5 and the brightness of the night sky. The effect of declaration of a state of emergency on the brightness of night sky was not found in this analysis.

目次

第1章 序論	2
1.1 夜空の明るさ	2
1.2 岡山大学天文台スカイモニター	3
第2章 データ	4
2.1 データの選別	4
2.1.1 ゲイン	4
2.1.2 太陽光と月光	4
2.1.3 雲	5
2.2 データ選別の結果	6
2.3 夜空の明るさ	7
第3章 結果	8
3.1 季節変化	8
3.1.1 夜空の明るさ	8
3.1.2 PM2.5と夜空の明るさ	12
3.2 緊急事態宣言の影響	13
第4章 まとめ	15
謝辞	16
参考文献	17
付録	18

第1章 序論

1.1 夜空の明るさ

夜空の明るさは、日により、時間により違う。太陽が沈んだ直後や月が出ていたりすると夜空は明るくなるが、それ以外にも空の明るさは変わる。図 1-1 は岡山大学天文台に設置されたスカイモニターが撮影した画像である。中心は天頂で、上が北、下が南になっている。白く写っているのは雲で、南方が明るくなっているのは街明かりの影響である。

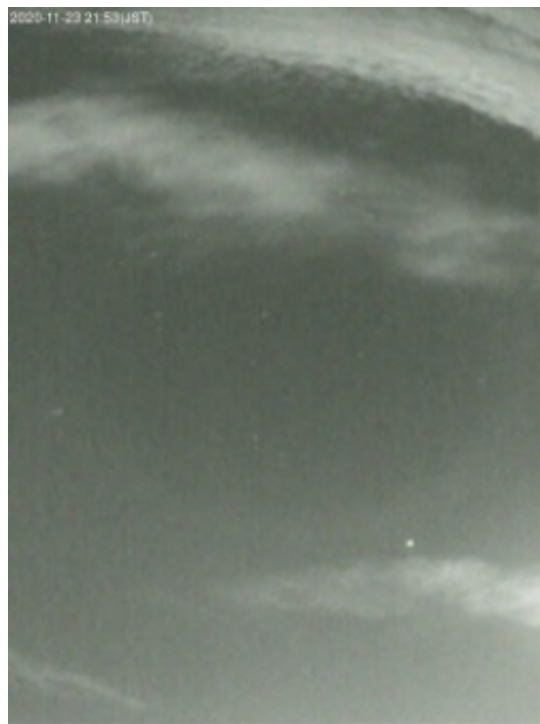


図 1.1: 岡山大学天文台スカイモニターで撮影された画像。

福本 (2022) は、2019 年 7 月 6 日～2021 年 12 月 31 日に岡山天文台スカイモニターが撮影した画像を用いて、視野に月や雲がない晴れた夜の空の明るさの変化と季節変化について調べた。1 晩の明るさの変化については、18 時から 24 時の前半夜で徐々に暗くなり、24 時から 30 時 (翌日の 6 時) の後半夜ではあまり変わらない事が明らかとなった。また、前半夜は毎正時に明るさが急激に暗くなるという

現象が見られたことから、夜空の明るさの変化が人間活動の影響を強く受けていることが示唆された。季節変化については春に明るく冬に暗いという傾向があるように見えたが、期間内で測定できた夜が29夜しかなく、データ数の少なさから季節変化を定量的に評価することはできなかった。

本研究は、岡山大学天文台スカイモニターが撮影した画像を用いて、夜空の明るさの季節変化を定量的に評価した。解析期間を福本(2022)の約2年半から4年半に延ばすとともに、解析手法を工夫することで、より多くの夜について空の明るさを測定した。

1.2 岡山大学天文台スカイモニター

岡山大学天文台スカイモニターの詳細は福本(2022)に記述されている。スカイモニターは岡山大学津島キャンパスにある7階建の自然科学研究科棟の屋上に設置され2016年から1分ごとに24時間、定常的に空の撮影をしている。撮影には市販の広角屋外用防犯カメラを使用している(表1.1)。このカメラは水平115度垂直65度の超広角の視野を1920x1080の解像度で撮影する。明るい昼間から暗い夜間まで撮影するため、周囲の明るさに合わせてゲインの調整がおこなわれる。また、明るいときは24bitカラーで撮影するが、暗いときは感度を上げるため8bit白黒に切り替えて撮影される。ゲイン調整とカラー/白黒の切り替えは、自動的におこなわれる。カメラの出力はビデオキャプチャアダプタによって640x480の解像度で取り込まれ、1分毎に保存される。

表 1.1: スカイモニターが使用しているカメラ(型番:SJC21B)の諸元(福本(2022)から引用)

イメージセンサー	248万画素 1/2.8インチ SONY Exmor CMOS イメージセンサー
画角	115度 x65度
出力解像度	FHD(1920x1080)
最低被写体照度	白黒: 0.0008lux

出典: <https://www.ykmusen.co.jp/cctv/okugaicamera/sjc21bcvbs.html>

第2章 データ

解析には、スカイモニターが同一の機材・同一の設定で撮影した、2019年7月6日から2023年12月10日の期間の画像を用いた。

2.1 データの選別

スカイモニターの撮影は一部欠測はあるものの、おおよそ1分間隔で行われており、期間中に撮影された画像の枚数は計1143761枚に上る。人工光と大気エアロゾルの影響を評価するため、人工光と大気エアロゾル以外が夜空の明るさに影響していると考えられる画像を、解析から除外した。本研究では、画像から夜空の明るさを求めるにあたって、(1)ゲイン、(2)太陽光、(3)月光、(4)雲、以上4つの観点で画像の選別をおこなった。

2.1.1 ゲイン

前章で述べたようにスカイモニターは周囲の明るさに合わせて自動的にゲインの値を変更して撮影する。異なるゲインで撮影された画像は、そのままでは明るさの比較ができないため、ゲイン最大で撮影された画像だけを使用することにした。画像の平均輝度値の時間変化からゲイン切替のタイミングを特定して、ゲインが最大でないと判断した画像を除外した。

2.1.2 太陽光と月光

太陽光と月光が夜空の明るさに及ぼす影響を取り除くため、日の出・日の入り、月の出・月の入りの時刻を「各地のこよみ」(国立天文台)から入手し、太陽光については薄明の影響を考慮して日の入りから90分後までと、日の出の90分前以降に撮影されたデータは除外し、月光については月の出から月の入りまでの間に撮影されたデータを除外した。

2.1.3 雲

画像の切り抜き

福本(2022)では、1晩中ずっと雲がないという条件で画像を選別した。その場合、図2.1のように視野の一部に雲は出ているが、晴れている部分もあるという画像は全て破棄されていた。本研究では図のように画像の中心部分を切り出して、画像全体でなく天頂付近(画角13度四方となる範囲)だけの明るさを測定することにした。天頂付近のみを切り出すことで、視野の一部に雲がかかっている画像からも、天頂部分が晴れていれば雲のない空の明るさが測定されるようになった。この画像の切り抜きによって、本研究は福本(2022)に比べて大幅に測定数を増やすことに成功した。画像の切り抜きは、ビデオキャプチャアダプタで取り込む際に生じた縦横比のズレを修正してからおこなった。



図 2.1: 画像の切り抜きの模式図。右はビデオキャプチャで取り込む際に生じた縦横比を修正した後の画像。

雲の自動判定

100万枚以上の画像を目で見て判定するわけにはいかないため、雲の有無を自動判定するアルゴリズムを作って、自動判定によって雲のない画像を選別した。判定には1分間隔で撮影された2枚の画像の差分を用いた。雲は風に流されて移動するため、図のように00:00から00:01にかけて視野内での位置が変わる。この2枚の画像の差分を取ると図のようになる。この差分の画像の平均輝度値が大きいと雲があると判定し、画像を測定から除外した。

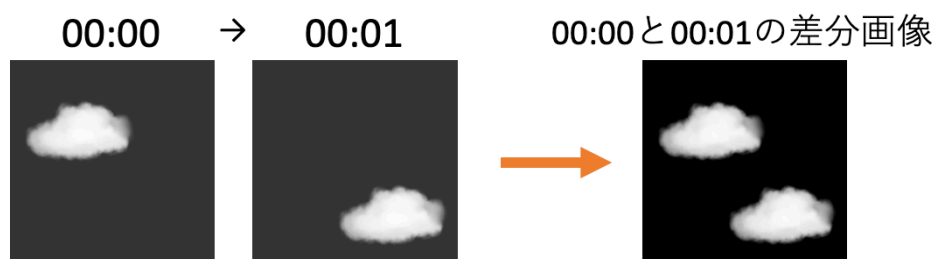


図 2.2: 差分画像の模式図

2.2 データ選別の結果

データ選別の結果の一例を示す。図は2020年1月29日のデータで、横軸が時間で左端が18時、右端が翌朝の6時である。縦軸は各時刻に撮影された画像の平均輝度値で上に行くほど明るくなる。図の緑の範囲は薄明の時間、黄色の範囲は月が出ていた時間、各点は各時刻に撮影された画像から求めた夜空の明るさで、灰色は選別で除外されたデータ、赤色は除外されなかったデータである。緑色や黄色のハッチがかかっていないもので除外されているものは「雲あり」と判定されたものである。一部、雲の影響があるように見えないデータも「雲あり」と判定されて除外されているが、多くの「雲なし」データを抽出することに成功した。この夜のデータは25時半から27時に雲が出ているため福本(2020)では全データを捨てていたが、本研究では雲の自動判定を導入したことで、雲が出た時間以外のデータを使うことができるようになった。

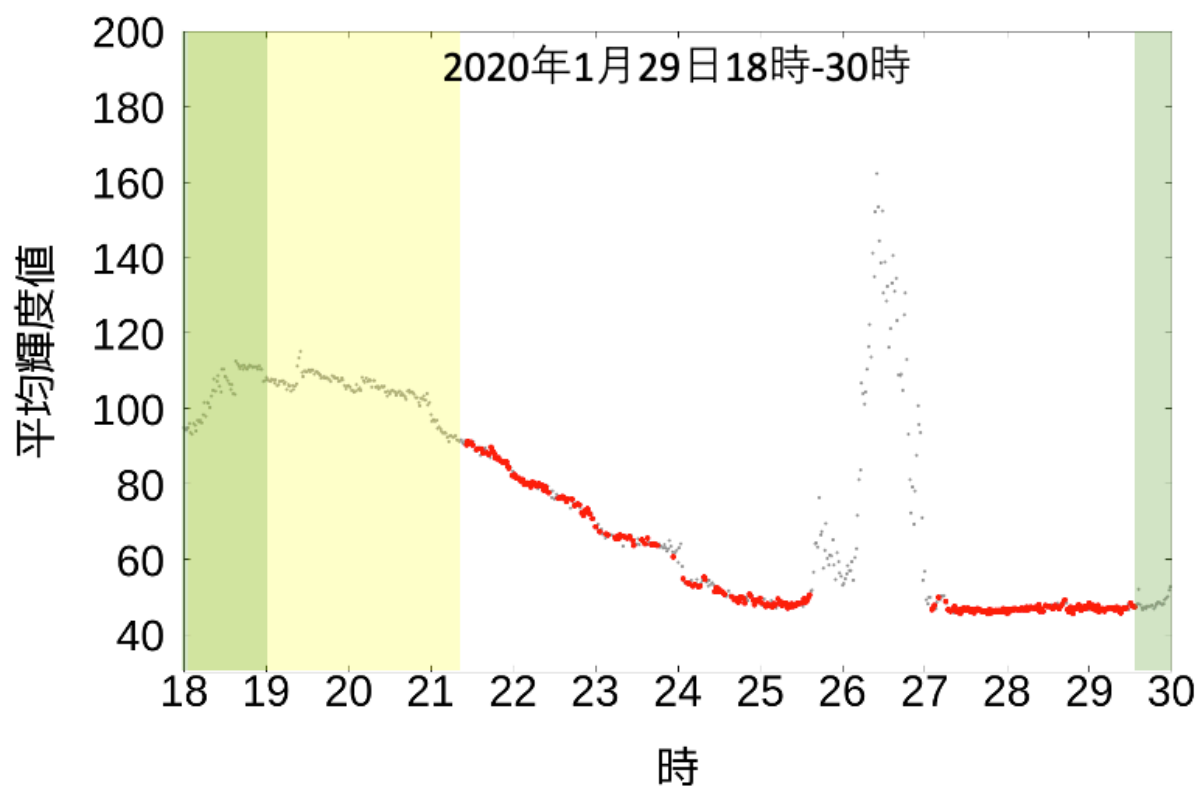


図 2.3: データ選別の結果の一例. 2020年1月29日

2.3 夜空の明るさ

選別で除外されなかった画像は、各画像ごとに平均輝度値を求め、その値をその時刻における夜空の明るさとした。画像は天頂付近を切り抜いているので、ここで求めた夜空の明るさは天頂付近の明るさということになる。

第3章 結果

3.1 季節変化

3.1.1 夜空の明るさ

街明かりの影響を除くため、後半夜の25時から28時に撮影された画像だけを用いて夜空の明るさを評価した。終わりの時間を28時としたのは季節によって日の出の時刻が変わることの影響を避けるためである。各夜について、25時から28時の間に測定された各時刻の夜空の明るさの平均を計算し、計算された平均値をその夜の明るさとした。図3.1は約4年半の期間に測定された夜の明るさを全てプロットしたもので、横軸は日で左端が1月1日、右端が12月31日となっている。図の黒い点は各夜の明るさで、卒論では29夜のみあったものが、本研究では288夜、卒業研究の約10倍の夜がプロットされている。赤で描いたのは各月の平均で、エラーバーは95%信頼区間を表している。卒業研究ではデータ数が少なく、測定の誤差か、ゆらぎか、統計的に有意な季節変化があるのかどうかわからなかったが、今回はデータの数が増えたため、季節によって夜空の明るさが有意に異なることが明らかになった。3-5月の春頃と8月~1月の秋から冬は、夜空の明るさの平均がはっきりと異なり、岡山の夜空の明るさには明瞭な季節変化があることが示された。

図3.2から3.6の各図は、各年ごとにデータを分けた場合である。いずれの年も、全てまとめた場合(図3.1)と似たような季節変化をしている。年によって違いがあるかどうか議論したいところだが、各年ごとにデータを分けるとデータ数が少なくなり区間推定の幅が広がってしまうため、年による違いを議論することは難しい。

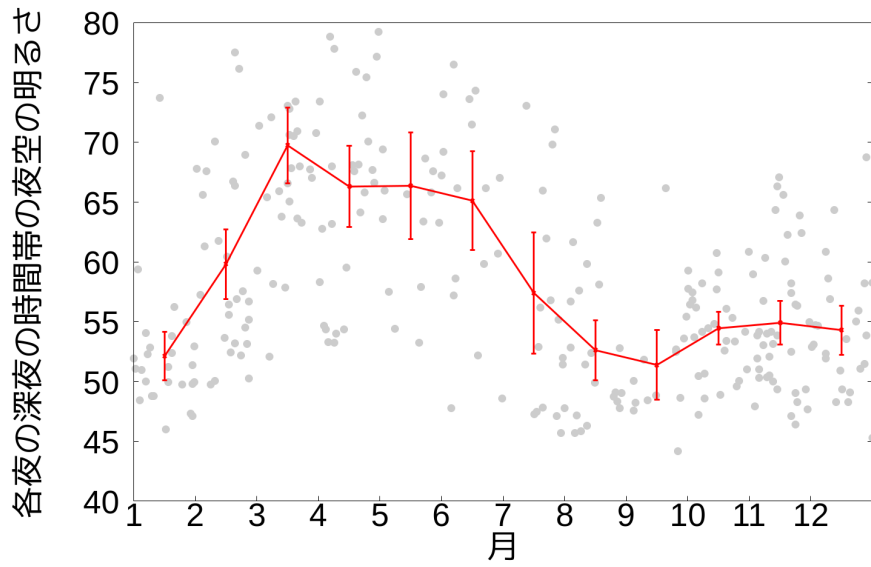


図 3.1: 夜空の明るさの季節変化 (2019 年から 2023 年の全データ).

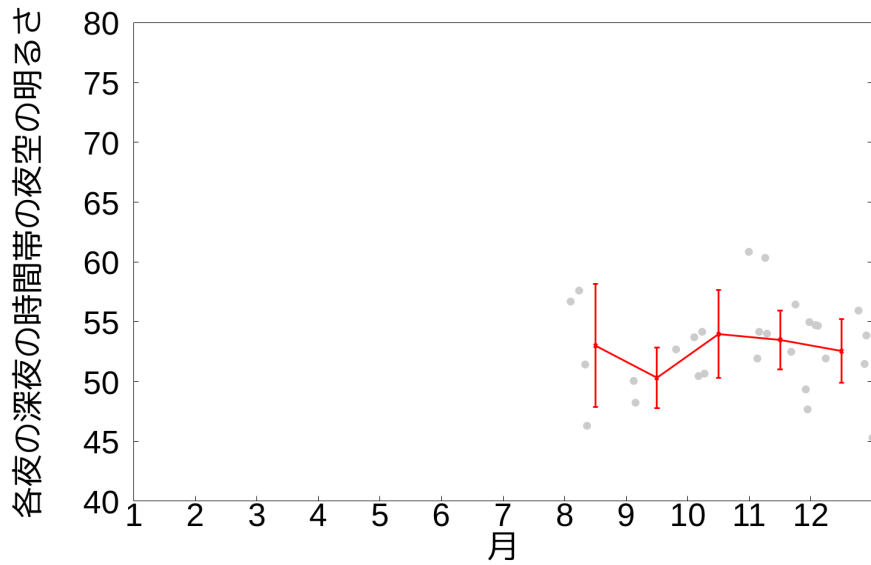


図 3.2: 夜空の明るさの季節変化 (2019 年).

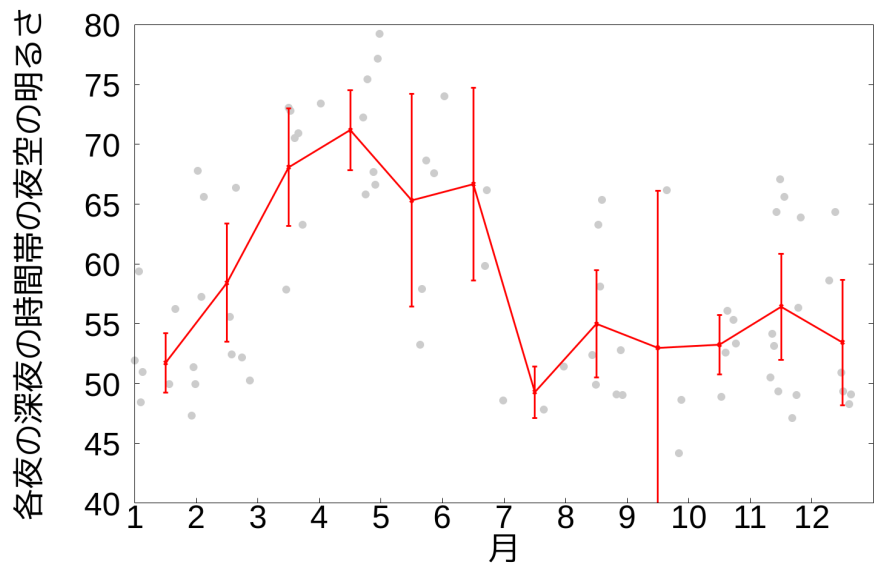


図 3.3: 夜空の明るさの季節変化 (2020 年).

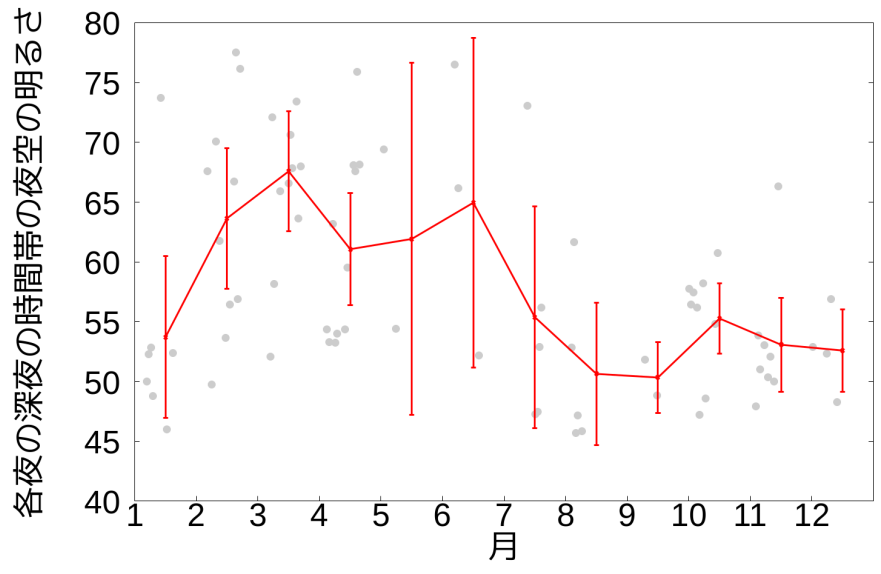


図 3.4: 夜空の明るさの季節変化 (2021 年).

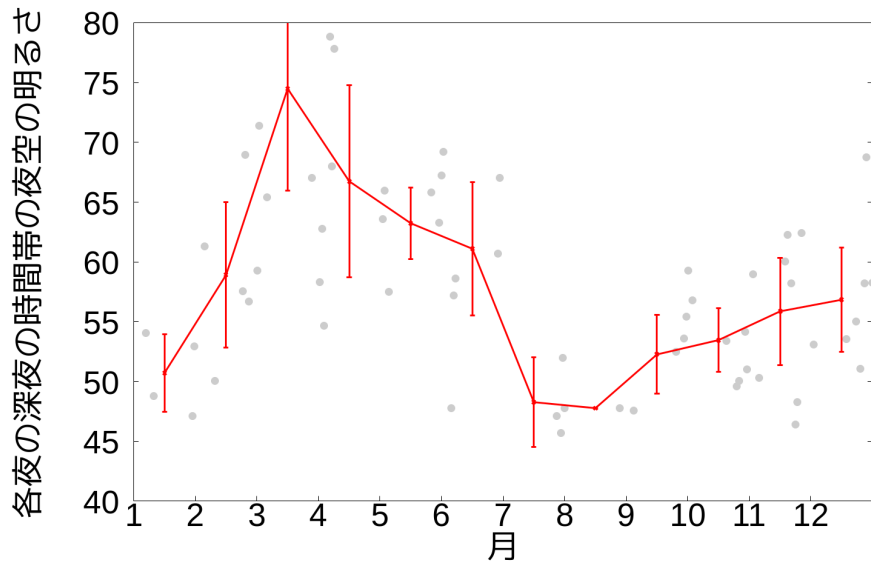


図 3.5: 夜空の明るさの季節変化 (2022 年).

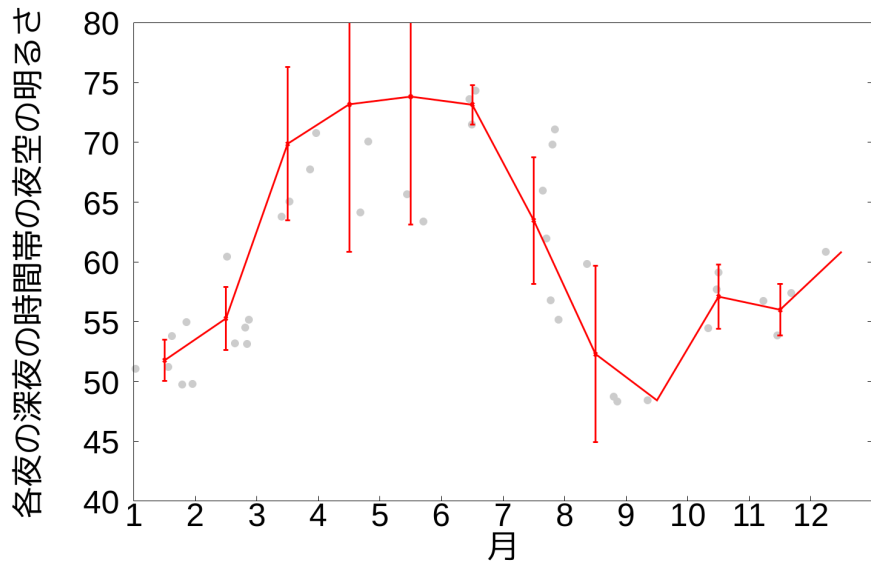


図 3.6: 夜空の明るさの季節変化 (2023 年).

3.1.2 PM2.5 と夜空の明るさ

何もない夜空では、大気中に浮かぶエアロゾルによる光の散乱が夜空の明るさに影響していると考えられる。図3.7は、岡山大学スカイモニターで測定した夜空の明るさの月平均と、岡山環境保健センターが吉備で測定したPM2.5の月平均である。岡山大学スカイモニターで夜空の明るさの月平均が測定できたのは2019年8月から2023年12月までだが、岡山環境保健センターのデータは2019年1月から2022年12月までだけが入手可能だったので、図3.7には全部で41個の点がプロットされている。図の赤色の実線は回帰直線である。図からわかる通りPM2.5と夜空の明るさには正の相関がある。相関があるからといって因果関係があるとは言えないが、PM2.5の量が夜空の明るさを決定している可能性が高いと考えられる。

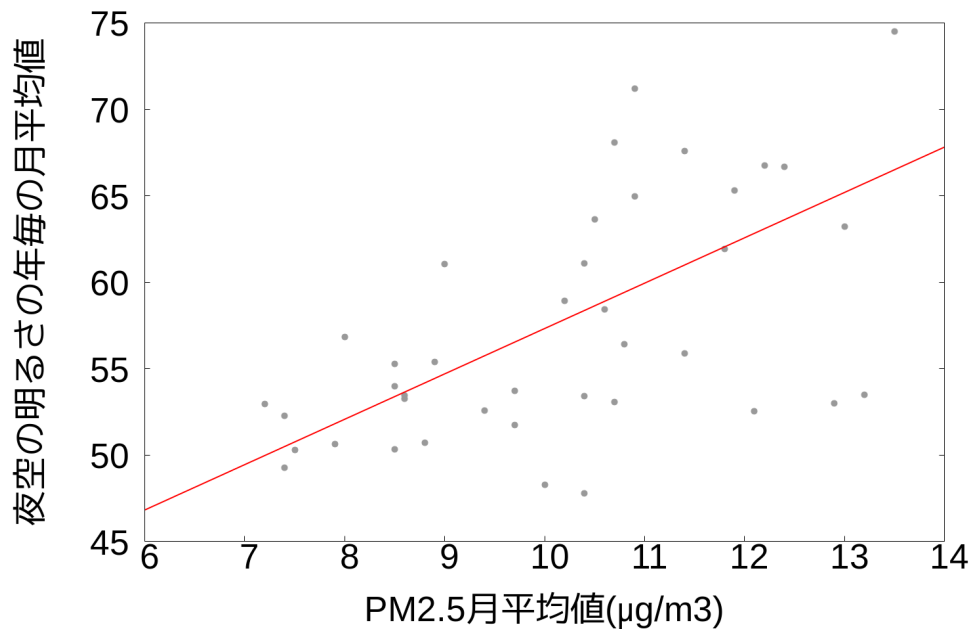


図 3.7: PM2.5 と夜空の明るさ

3.2 緊急事態宣言の影響

図3.8-3.10は、過去3回岡山で緊急事態宣言が発令されていた期間について、21時から28時(翌日の4時)の各時刻の夜空の明るさを図にしたものである。緊急事態宣言のあった年は赤色、それ以外の年は灰色でデータをプロットしている。緊急事態宣言の発出中(赤)とそれ以外(灰)に差があるようには見えず、緊急事態宣言の発出は夜空の明るさに影響しなかったように見える。

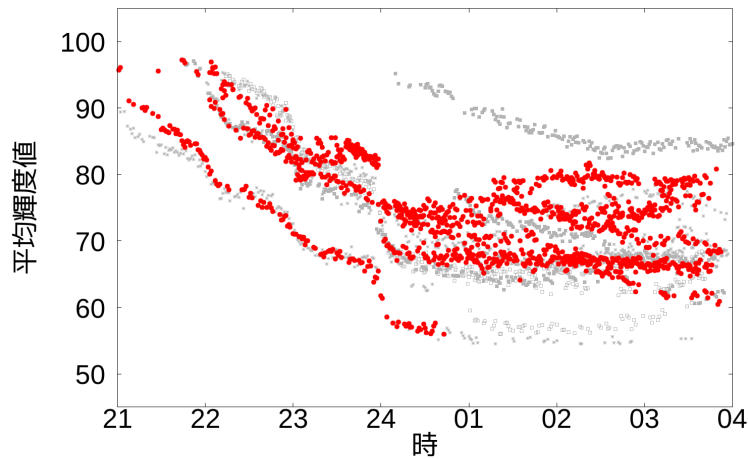


図 3.8: 4月16日-5月14日(2020-2023)の夜空の明るさ。2020年の同期間に緊急事態宣言が発出されていた。

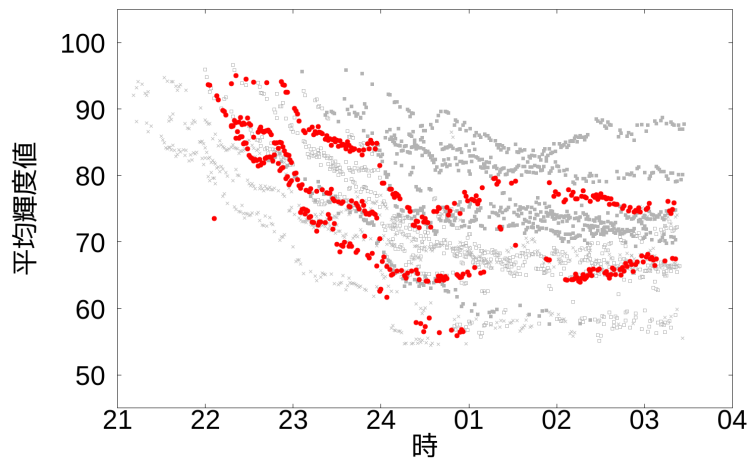


図 3.9: 5月16日-6月20日(2020-2023)の夜空の明るさ。2021年の同期間に緊急事態宣言が発出されていた。

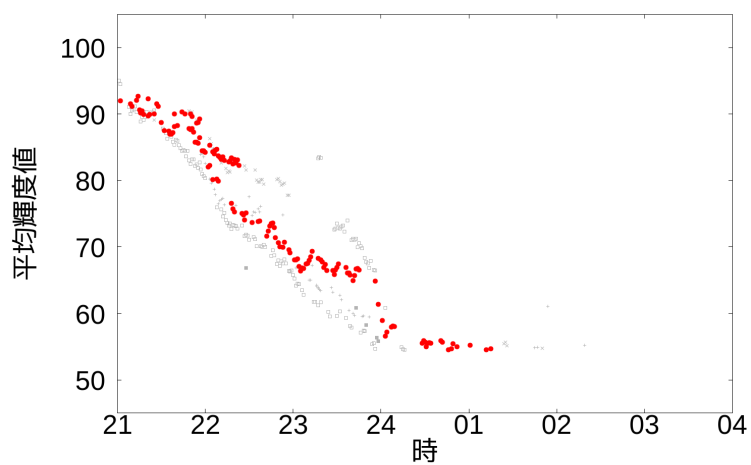


図 3.10: 8月27日-9月11日 (2019-2023) の夜空の明るさ. 2021年の同期間に緊急事態宣言が発出されていた.

第4章 まとめ

岡山大学天文台スカイモニターが撮影した画像を用いて夜空の明るさを測定した。2019年7月から2023年12月の約4年半の期間から288夜について夜空の明るさを測定したことで、岡山の夜空の明るさは3月から5月の春に明るく、8月から1月の秋から冬に暗いという明瞭な季節変化があることが明らかになった。またPM2.5と夜空の明るさには正の相関があることを示した。緊急事態宣言の夜空の明るさへの影響については、今回の解析結果からは影響は見えなかった。

謝辞

本研究を行うにあたり、ご指導いただきました指導教員である、はしもとじょーじ教授に心より感謝いたします。

参考文献

福本佑典 (2022) :

岡山大学天文台スカイモニターを用いた夜空の明るさに関する研究
岡山大学理学部地球科学科 2021 年度 課題研究.

岡山県環境保健センター :

環境大気の常時監視測定結果 (月間値), 2023-11-21 更新
<https://www.pref.okayama.jp/site/712/798171.html>

国立天文台 :

各地のこよみ

<https://eco.mtk.nao.ac.jp/koyomi/dni/>

付録

本研究では(1)画像の処理, (2)雲の自動判定, (3)季節変化, (4)緊急事態宣言の影響について作業を行った. 以下, それぞれで用いたソースコードについて説明する

画像の処理

画像の処理では画像の切り抜き, 差分画像の生成, 画像の平均輝度地のテキストファイルにする作業を行いました.

画像の切り抜きには, 以下のソースコードを利用した.

ソースコード 1: crop.sh

```
1
2 #!/bin/bash
3
4 YYYY=年 (2019-2023)
5 MM=printf で 2 桁にした月 (01-12)
6 DD=printf で 2 桁にした日 (01-月末)
7 HH=printf で 2 桁にした時 (18-23,00-05)
8
9 CONVERT=/usr/bin/convert
10
11 #分の指定
12 for i in `seq 0 59`; do
13
14 #分を 2桁にする (00-59)
15     mm=$( printf "%02d" $i )
16
17     FILE=処理するスカイモニターの画像の絶対パス
18
19 #ファイルがある場合
20     if [ -e $FILE ]; then
21
22 #切り抜き前画像の中心 96x72pixel を切り抜き
23 #72x72 のサイズにして切り抜き後画像にする
24     $CONVERT -gravity center -crop 96x72+0+0 -resize 72x72!
        切り抜き前画像 切り抜き後画像
```

岡山大学天文台スカイモニターを用いた夜空の明るさの季節変化に関する研究19

25 fi
26
27 done

差分画像の生成には、以下のソースコードを利用した。

ソースコード 2: sabuncrop.sh

```
1
2 #!/bin/bash
3
4 #月末の日の設定
5 mm[1]=31
6 mm[2]=28
7 mm[3]=31
8 mm[4]=30
9 mm[5]=31
10 mm[6]=30
11 mm[7]=31
12 mm[8]=31
13 mm[9]=30
14 mm[10]=31
15 mm[11]=30
16 mm[12]=31
17
18
19 YYYY= 年 (2019-2023)
20 M= 月 (1-12)
21 D= 日 (1-月末)
22 MM= printf で 2 桁にした月 (01-12)
23 DD= printf で 2 桁にした日 (01-月末)
24 HH= printf で 2 桁にした時 (18-23,00-05)
25 sabun= 何分間隔の差分画像を作るかの指定、本研究では 1 を入れた
26
27 #閏年の月末の処理
28 if [ ${YYYY} -eq 2020 ]; then
29     mm[2]=29
30 else
31     mm[2]=28
32 fi
33
34 COMPOSITE=/usr/bin/composite
35
36 pass1= 処理する画像を置いてあるディレクトリの絶対パス
37 pass2= 差分画像を生成するディレクトリの絶対パス
38
39 #23時 59分から 00時 00分の切り替えの処理
40 #処理している日が月末でなければ
41 if [ $D -ne ${mm[$M]} ]; then
42
43 #日付を 1日進める
```

岡山大学天文台スカイモニターを用いた夜空の明るさの季節変化に関する研究21

```
44     DDD=$( printf "\%02d" $(( $D + 1 )) )
45     MMM=$MM
46     YYYY2=$1
47
48 #処理している日が月末で
49 else
50
51 #なおかつ 12月以外なら
52     if [ $M -ne 12 ]; then
53
54 #日と月を 1進める
55         DDD="01"
56         MMM=$( printf "\%02d" $(( $M + 1 )) )
57         YYYY2=$1
58
59 #12月なら
60     else
61
62 #日と月と年を 1進める
63         DDD="01"
64         MMM="01"
65         YYYY2=$(( ${YYYY} + 1 ))
66
67     fi
68 fi
69
70
71 pass3= 日付を超えた後の差分画像を生成するディレクトリへの絶対パス
72
73
74 #00分~(59-sabun)分の差分画像の作成
75
76 end=$(( 59 - ${sabun} ))
77
78 for i in `seq 0 ${end}`; do
79
80 #変数i を 2桁で表しそれを mm に入れる
81     mm=$( printf "\%02d" $i )
82 #変数i+1を 2桁で表しそれをmmmに入れる
83     mmm=$( printf "\%02d" $(( $i + 1 )) )
84
85 #差分画像を作成する
86     $COMPOSITE 画像A 画像の 1分後に撮影された画像B -compose
87         difference 差分画像 2> /dev/nul
88 done
```

```
89
90
91 # (60-sabun)分~00分の差分画像の作成
92
93 #23時以外なら
94 if [ ${HH} -ne 23 ]; then
95
96
97     for o in ` seq 1 ${sabun} `; do
98
99         n=$(( $o - 1 ))
100        q=$(( ${sabun} - ${n} - 1 ))
101
102        #時を1進める
103
104        mm=$(( 59 - $q ))
105        mmm=$( printf "%02d" $n )
106        HHH=$( printf "%02d"=$(( $4 + 1 )) )
107
108
109        #差分画像を作成する
110        $COMPOSITE 画像A 画像の1分後に撮影された画像B -compose
        difference 差分画像 2> /dev/nul
111
112        done
113
114        #23時なら
115        else
116
117        for o in ` seq 1 ${sabun} `; do
118
119            n=$(( $o - 1 ))
120            q=$(( ${sabun} - ${n} - 1 ))
121
122            #時は0にする
123            mm=$(( 59 - $q ))
124            mmm=$( printf "%02d" $n )
125            HHH=00
126
127            #差分画像を作成する
128            $COMPOSITE 画像A 画像の1分後に撮影された画像B -compose
            difference 差分画像 2> /dev/nul
129
130            done
131
132        fi
```

画像の平均輝度地のテキストファイル化には、以下のソースコードを利用した。

ソースコード 3: txt.sh

```
1
2 #!/bin/bash
3
4 YYYY=年 (2019-2023)
5 MM=printf で幅を 2 にした月 (01-12)
6 DD=printf で幅を 2 にした日 (01-月末)
7 HH=printf で幅を 2 にした時 (18-23,00-05)
8
9 IDENTIFY=/usr/bin/identify
10
11 pass=処理する画像のあるディレクトリの絶対パス
12 pass1=テキストファイルを生成するディレクトリの絶対パス
13
14 for i in `seq 0 59`; do
15     mm=$( printf "%02d" $i )
16     FILE=画像の絶対パス
17
18 #画像があれば画像の平均輝度値をテキストに書き込む
19     if [ -e $FILE ]; then
20         $IDENTIFY -format '\%[mean], \f\n' 画像 >> テキスト
21
22 #画像がなければNaNを記入する
23     else
24         nitiji="NaN , ${YYYY}${MM}${DD}${HH}${mm}crop.png"
25         echo "$nitiji" >> テキスト
26     fi
27
28 done
```

雲の自動判定

雲の自動判定は以下のソースコードで実行した。

ソースコード 4: hantei.sh

```
1
2 #!/bin/bash
3
4 #区切り文字を/と,としてspike4.csvの1,2,3列目を切り取る
5 #spike4.csvの1,2,3列目にはそれぞれ年,月,日が記述されている
6 #1行ずつ読み取る、1列目には
   YYYY、2列目はMMM、3列目はDDDという変数とする
7
8 awk -F "[/,]" '{print $1,$2,$3}' /home/fuku/work/data/spike4.csv
   | while read YYYY MMM DDD ; do
9
10 #変数MMM・DDDは2桁で表しそれぞれMM・DDに入れる
11   MM=$( printf "%02d" $MMM )
12   DD=$( printf "%02d" $DDD )
13
14
15   kidoikiti= 輝度の閾値を設定するため600を入れる
16
17 #判定結果を出力するディレクトリを生成する
18   mkdir /home/fuku/work/data/hantei/k${kidoikiti} 2>/dev/null
19   mkdir /home/fuku/work/data/hantei/k${kidoikiti}/${YYYY} 2>/
   dev/null
20   mkdir /home/fuku/work/data/hantei/k${kidoikiti}/${YYYY}/${
   YYYY}${MM} 2>/dev/null
21
22 #区切り文字を,として
23 #差分画像の平均輝度値を記載したテキストファイルの1,2列目を切り取る
24 #差分画像の平均輝度値を記載したテキストファイルの1,2列目には
25 #輝度値と画像のファイル名が記述されている
26 #1行ずつ読み取る、1列目はkido、2列目はfileという変数とする
27
28   awk -F " , " '{print $1,$2}' sabun.txt | while read kido file;
   do
29
30
31 #kidoがNaNであれば
32   if [ ${kido} == NaN ]; then
33
34 #判定結果のテキストファイルにはNaNを入れる
35   echo "${file},NaN" >> /home/fuku/work/data/hantei/k${
   kidoikiti}/${YYYY}/${YYYY}${MM}/${YYYY}${MM}${DD}
```

```
hantei.txt
36
37 #kido が NaN でなければ
38     else
39
40 #閾値以下なら
41     if [ 'echo "${kido} < ${kidoikiti}" | bc' == 1 ];
42     then
43 #1を判定結果のテキストファイルに入れる
44     echo "${file},1" >> /home/fuku/work/data/hantei/
45         k${kidoikiti}/${YYYY}/${YYYY}${MM}/${YYYY}${MM}
46         ${DD}hantei.txt
47
48 #閾値以上なら
49     else
50 #0を判定結果のテキストファイルに入れる
51     echo "${file},0" >> /home/fuku/work/data/hantei/
52         k${kidoikiti}/${YYYY}/${YYYY}${MM}/${YYYY}${MM}
53         ${DD}hantei.txt
54
55 fi
56
57 done
58
59 echo ${YYYY}${MM}${DD}
60
61 done
```

季節変化

季節変化では雲がない、太陽・月の影響がない、ゲイン最大、品質管理で問題がないとされたデータを集めたテキストファイルの生成、と晩毎の夜空の明るさの平均と区間推定を行った

雲がない、太陽・月の影響がない、ゲイン最大、品質管理で問題がないとされたデータを集めたテキストファイルの生成は以下のソースコードで行った。

ソースコード 5: season.sh

```
1
2 #!/bin/bash
3
4 YYYY= 年 (2019-2023)
5 M= 月 (1-12)
6 D= 日 (1-月末)
7 MM= printf で 2 桁にした月 (01-12)
8 DD= printf で 2 桁にした日 (01-月末)
9 day=1月 1日から処理する日までの日数
10 pass=処理するテキストデータが置かれているディレクトリの絶対パス
11 file=処理するテキストデータの絶対パス
12
13 #計算に用いる変数にあらかじめ 0 を入れておく
14 kousuu=0
15 sum=0
16 ave=0
17
18 #処理するテキストデータがあれば
19 if [ -e "${file}" ]; then
20
21 #輝度値,日時,雲,ゲイン,太陽,月,品質管理
22 #の結果をまとめたテキストデータの
23 #25時-28時のデータを一行ずつ読み込んで
24 #平均輝度値をkido、ゲインをgain、雲の有無をcloud、
25 #太陽の有無をsun、月の有無をmoon
26 #品質管理の結果をvalidの変数に入れる
27
28     while read kido gain cloud sun moon valid; do
29
30 #kido,gain,cloud,sun,moon すべて NaN 出ないとき
31     if [ ${kido} != NaN ] && [ ${gain} != NaN ] && [ ${cloud}
32         } != NaN ] && [ ${sun} != NaN ] && [ ${moon} != NaN
33         ]; then
34
35         total=0
```

```
35
36 #cloud,gain.sun.validを合計してtotalという変数に入れる
37 #利用可能な時刻のtotalは4になる
38     total=$(( {cloud} + {gain} + {sun} + {valid} ))
39
40 #totalが4かつ月が出ていない時
41     if [ {total} -eq 4 ] && [ {moon} == 0.000 ]; then
42
43 #利用可能なデータの数をkousuuという変数に入れて数える
44 #利用可能な時刻の平均季節変化をsumという変数に入れて合計する
45     kousuu=$(( {kousuu} + 1 ))
46     sum=$( bc <<< "scale=2;{sum} + {kido} " )
47
48     fi
49     fi
50 done <<(cat {pass}/{file} | head -n 600 | tail -n 180 | awk
51     -F'[ , ]' '{print $1,$10,$12,$13,$14,$15}' )
52
53 #利用可能なデータ数が0より多ければ
54     if [ {kousuu} -gt 0 ]; then
55
56 #sumとkousuuから平均を求めてaveという変数に入れる
57     ave=$( bc <<< "scale=2; {sum} / {kousuu} " )
58
59 #aveが25000より小さければ
60 #平均、年、月、日、1月1日からの日数、データ数をテキストデータに記入す
61     る
62     if [ 'echo "{ave} < 25000" | bc' == 1 ]; then
63         echo "{ave},{YYYY},{MM},{DD},{day},{kousuu}" >>
64             /home/fuku/work/data/valid7/{YYYY}valid.txt
65         echo "{ave},{YYYY},{MM},{DD},{day},{kousuu}" >>
66             /home/fuku/work/data/valid7/valid.txt
67     fi
68     fi
69     fi
70 fi
```

晩毎の夜空の明るさの平均と月毎の区間推定は以下のソーススクリプトで計算した

ソースコード 6: valid.sh

```
1
2 #!/bin/bash
3
4 #1月～12月について
5 for M in `seq 1 12`; do
6
7 #後で出てくる変数にあらかじめ0を入れておく
8     sum1=0
9     sum2=0
10    kousuu1=0
11    kousuu2=0
12    ave=0
13    bun=0
14    s=0
15    se=0
16
17 #season.sh で作成したテキストデータを一行ずつ読み込んで
18 #平均輝度値をkido, 月をMMという変数に入れる
19     while read kido MM; do
20
21 #読み込んだ月MMと処理している月Mが等しければ
22     if [ ${MM} -eq ${M} ]; then
23
24 #平均輝度値のsum1 という変数で合計する
25         sum1=$( bc <<< "scale=3; ${sum1} + ${kido} ")
26
27 #データ数をkousuu1 という変数で数える
28         kousuu1=$(( ${kousuu1} + 1 ))
29     fi
30
31     done <<(cat /home/fuku/work/data/valid7/valid.txt | awk -F
32         ',,' '{print $1,$3}' )
33
34 #平均を計算してave という変数に入れる
35
36 #season.sh で作成したテキストデータを一行ずつ読み込んで
37 #平均輝度値をkido, 月をMMという変数に入れる
38     while read kido MM; do
39
40 #読み込んだ月MMと処理している月Mが等しければ
41     if [ ${MM} -eq ${M} ]; then
```

```
42
43 #kido と ave を用いて分散を求めて bun に入れる
44     bun=$( bc <<< "scale=3; (${kido} - ${ave})^2 ")
45
46 #分散の合計を計算してsum2に入れる
47     sum2=$( bc <<< "scale=3; ${sum2} + ${bun} ")
48     fi
49
50 done <<(cat /home/fuku/work/data/valid7/valid.txt | awk -F
    ', ' '{print $1,$3}' )
51
52
53     kousuu2=$(( ${kousuu1} - 1 ))
54     s=$( bc <<< "scale=3; ${sum2} / ${kousuu2} ")
55
56 #標準偏差を求める
57     sq=$( bc <<< "scale=3; sqrt(${s})")
58
59 #標準誤差を求める
60     se=$( bc <<< "scale=3; sqrt(${s} / ${kousuu1}) ")
61
62 #95%信頼区間を求める
63     low=$(bc <<< "scale=3; ${ave} - ( 1.96 * ${se} )")
64     high=$(bc <<< "scale=3; ${ave} + ( 1.96 * ${se} )")
65
66 #月、平均、信頼区間をテキストファイルに書き込む
67     echo "${M},${ave},${low},${high}" >> /home/fuku/work/data/
        valid7/allestimation.txt
68
69     echo $M
70
71 done
```

緊急事態宣言の影響

緊急事態宣言の影響では緊急事態宣言の期間ごとに雲がない、太陽・月の影響がない、ゲイン最大、品質管理で問題がないとされたデータを集めたテキストファイルの生成、期間内の同じ時間の平均の計算を行った。

緊急事態宣言の期間ごとに雲がない、太陽・月の影響がない、ゲイン最大、品質管理で問題がないとされたデータを集めたテキストファイルは以下のソースコードで生成した。

ソースコード 7: corona.sh

```
1
2 #!/bin/bash
3
4 YYYY= 年 (2019-2023)
5 M= 月 (1-12)
6 D= 日 (1-月末)
7 MM= printf で 2 桁にした月 (01-12)
8 DD= printf で 2 桁にした日 (01-月末)
9 day=1月 1日から処理する日までの日数
10 pass=処理するテキストデータが置かれているディレクトリの絶対パス
11 file=処理するテキストデータの絶対パス
12
13 #計算に用いる変数にあらかじめ0を入れておく
14 kousuu=0
15 sum=0
16 ave=0
17
18
19 #処理するテキストデータがあれば
20 if [ -e "${file}" ]; then
21
22 #輝度地,日時,雲,ゲイン,太陽,月,品質管理
23 #の結果をまとめたテキストデータの
24 #21時-24時のデータを一行ずつ読み込んで
25 #平均輝度値をkido、18時00分を1とした時の経過の分数をline、
26 #ゲインをgain、雲の有無をcloud、
27 #太陽の有無をsun、月の有無をmoon、
28 #品質管理の結果をvalidの変数に入れる
29
30
31     while read kido line HH gain cloud sun moon valid; do
32
33 #kido,gain,cloud.sun.moon すべてNaN 出ないとき
34     if [ ${kido} != NaN ] && [ ${gain} != NaN ] && [ ${cloud
        } != NaN ] && [ ${sun} != NaN ] && [ ${moon} != NaN
```

```
]; then
35
36
37 #繰り返しのためにtotalを0にする
38     total=0
39
40 #cloud,gain.sun.validを合計してtotalという変数に入れる
41 #利用可能な時刻のtotalは4になる
42     total=$(( ${cloud} + ${gain} + ${sun} + ${valid} ))
43
44 #totalが4かつ月が出ていない時
45     if [ ${total} -eq 4 ] && [ ${moon} == 0.000 ]; then
46
47 #kidoが14000より大きく25000より小さい時
48     if [ 'echo " ${kido} < 25000 " | bc' == 1 ] && [
49         'echo " ${kido} > 14000 " | bc' == 1 ]; then
50 #テキストに書き込む
51     echo "${kido},${YYYY},${M},${D},${HH},${line
52         }" >> /home/fuku/work/data/corona5/all.txt
53
54 #4月16-30日を緊急事態宣言1回目の期間として
55 #輝度値、年、月、日、時、分をテキストファイルを生成する
56     if [ ${M} -eq 4 ] && [ ${D} -ge 16 ] && [ ${
57         D} -le 30 ]; then
58         echo "${kido},${YYYY},${M},${D},${HH},${
59             line}" >> /home/fuku/work/data/corona5/
60             corona1.txt
61         echo "${kido},${YYYY},${M},${D},${HH},${
62             line}" >> /home/fuku/work/data/corona5/
63             ${YYYY}corona1.txt
64         echo "${kido},${YYYY},${M},${D},${HH},${
65             line}" >> /home/fuku/work/data/corona5/
66             coronaall.txt
67         echo "${kido},${YYYY},${M},${D},${HH},${
68             line}" >> /home/fuku/work/data/corona5/
69             ${YYYY}coronaall.txt
70     fi
71
72 #5月1-14日を緊急事態宣言1回目の期間として
73 #輝度値、年、月、日、時、分をテキストファイルを生成する
74     if [ ${M} -eq 5 ] && [ ${D} -ge 1 ] && [ ${D
75         } -le 14 ]; then
```

```
66     echo "${kido},${YYYY},${M},${D},${HH},${
        line}" >> /home/fuku/work/data/corona5/
        corona1.txt
67     echo "${kido},${YYYY},${M},${D},${HH},${
        line}" >> /home/fuku/work/data/corona5/
        ${YYYY}corona1.txt
68     echo "${kido},${YYYY},${M},${D},${HH},${
        line}" >> /home/fuku/work/data/corona5/
        coronaall.txt
69     echo "${kido},${YYYY},${M},${D},${HH},${
        line}" >> /home/fuku/work/data/corona5/
        ${YYYY}coronaall.txt
70         fi
71
72 #5月 16-31日を緊急事態宣言 2回目の期間として
73 #輝度値、年、月、日、時、分をテキストファイルを生成する
74         if [ ${M} -eq 5 ] && [ ${D} -ge 16 ] && [ ${
        D} -le 31 ]; then
75             echo "${kido},${YYYY},${M},${D},${HH},${
                line}" >> /home/fuku/work/data/corona5/
                corona2.txt
76             echo "${kido},${YYYY},${M},${D},${HH},${
                line}" >> /home/fuku/work/data/corona5/
                ${YYYY}corona2.txt
77             echo "${kido},${YYYY},${M},${D},${HH},${
                line}" >> /home/fuku/work/data/corona5/
                coronaall.txt
78             echo "${kido},${YYYY},${M},${D},${HH},${
                line}" >> /home/fuku/work/data/corona5/
                ${YYYY}coronaall.txt
79         fi
80
81 #6月 1-20日を緊急事態宣言 2回目の期間として
82 #輝度値、年、月、日、時、分をテキストファイルを生成する
83         if [ ${M} -eq 6 ] && [ ${D} -ge 1 ] && [ ${D
        } -le 20 ]; then
84             echo "${kido},${YYYY},${M},${D},${HH},${
                line}" >> /home/fuku/work/data/corona5/
                corona2.txt
85             echo "${kido},${YYYY},${M},${D},${HH},${
                line}" >> /home/fuku/work/data/corona5/
                ${YYYY}corona2.txt
86             echo "${kido},${YYYY},${M},${D},${HH},${
                line}" >> /home/fuku/work/data/corona5/
                coronaall.txt
87             echo "${kido},${YYYY},${M},${D},${HH},${
```

岡山大学天文台スカイモニターを用いた夜空の明るさの季節変化に関する研究33

```
line}" >> /home/fuku/work/data/corona5/
${YYYY}coronaall.txt
88         fi
89
90 #8月 27-31日を緊急事態宣言 3回目の期間として
91 #輝度値、年、月、日、時、分をテキストファイルを生成する
92         if [ ${M} -eq 8 ] && [ ${D} -ge 27 ] && [ ${
D} -le 31 ]; then
93             echo "${kido},${YYYY},${M},${D},${HH},${
line}" >> /home/fuku/work/data/corona5/
corona3.txt
94             echo "${kido},${YYYY},${M},${D},${HH},${
line}" >> /home/fuku/work/data/corona5/
${YYYY}corona3.txt
95             echo "${kido},${YYYY},${M},${D},${HH},${
line}" >> /home/fuku/work/data/corona5/
coronaall.txt
96             echo "${kido},${YYYY},${M},${D},${HH},${
line}" >> /home/fuku/work/data/corona5/
${YYYY}coronaall.txt
97         fi
98
99 #9月 1-11日を緊急事態宣言 3回目の期間として
100 #輝度値、年、月、日、時、分をテキストファイルを生成する
101         if [ ${M} -eq 9 ] && [ ${D} -ge 1 ] && [ ${D
} -le 11 ]; then
102             echo "${kido},${YYYY},${M},${D},${HH},${
line}" >> /home/fuku/work/data/corona5/
corona3.txt
103             echo "${kido},${YYYY},${M},${D},${HH},${
line}" >> /home/fuku/work/data/corona5/
${YYYY}corona3.txt
104             echo "${kido},${YYYY},${M},${D},${HH},${
line}" >> /home/fuku/work/data/corona5/
coronaall.txt
105             echo "${kido},${YYYY},${M},${D},${HH},${
line}" >> /home/fuku/work/data/corona5/
${YYYY}coronaall.txt
106         fi
107     fi
108 fi
109 fi
110 done <<(cat ${pass}/${YYYY}${MM}${DD}matome.txt | head -n
360 | tail -n 180 | awk -F'[ , ]' '{print $1,$4,$8,$10,$12
,$13,$14,$15}' )
111
```


期間内の同じ時間の平均の計算は以下のソースコードで行った。

ソースコード 8: average.sh

```
1
2 #!/bin/bash
3
4 #2019年～2023年をYYYYという変数に順番に入れる
5 for YYYY in `seq 2019 2023 `; do
6
7 #緊急事態宣言が出ていた3つの期間を1,2,3として
8 #nonという変数に順番に入れる
9     for non in `seq 1 3`; do
10
11 #21時00分-23時59分の各時刻をline1という変数に順番に入れる
12     for line1 in `seq 181 360`; do
13
14
15 #後で出てくる変数にあらかじめ0を入れておく
16         sum1=0
17         sum2=0
18         kousuu1=0
19         kousuu2=0
20         ave=0
21         bun=0
22         s=0
23         se=0
24
25
26 #corona.shで作成されたテキストファイルを一行ずつ読み込んで
27 #平均輝度値はkidoという変数に入れる
28 #18時00分を1とした時の経過分数はline2という変数に入れる
29         while read kido line2; do
30
31 #テキストファイルから読み込まれた分と処理している分が同じ時
32             if [ ${line1} -eq ${line2} ]; then
33
34 #平均輝度値のsum1という変数で合計する
35                 sum1=$(( bc <<< "scale=3; ${sum1} + ${kido} "))
36
37 #データ数をkousuu1という変数で数える
38                 kousuu1=$(( ${kousuu1} + 1 ))
39
40             fi
41
42         done <<(cat /home/fuku/work/data/corona5/${YYYY}
43             corona${non}.txt | awk -F',' '{print $1,$6}' )
```

```
43
44 #平均を計算してave という変数に入れる
45     ave=$( bc <<< "scale=3; ${sum1} / ${kousuu1} ")
46
47 #line1 と平均を記述したテキストファイルを生成する
48     echo "${line1},${ave}" >> /home/fuku/work/data/
         corona5/${YYYY}corona${non}estimate.txt
49
50     echo ${YYYY}${M}
51
52     done
53 done
54 done
```
