

生活/クレジットカード/為替手数料

[Top](#) / [生活](#) / [クレジットカード](#) / [為替手数料](#)

- [VISA v. Master](#)
- [同日同店で競争](#)
- [TTM](#)
 - [自動化](#)
- [カード明細](#)
 - [Amex自動化](#)
- [Amexの意外な競争力](#)
- [このページのアーカイブ](#)

カードを外貨建てで使った場合の手数料（海外決済事務取扱手数料/Foreign transaction fees）は公式では1.63%の数字が多いが、実際の数字はなぜか違う。特にVISAとMasterの差が意外と大きい「感触がある」。「感触がある」ではなく、統計的に意味のある数字を出せる切り口は、個人の手持ちデータではかなり限られるが（縦横無尽の切り口でそれができるのはマネーフォワードなど限られた位置にある者だけ）、できる範囲で整理してみた。

VISA v. Master

ブランド	平均	誤差	信頼上限	信頼下限	標準誤差	観値
VISA	1.96%	±0.10	2.07%	1.86%	0.05	2.02%
Master	1.42%	±0.07	1.48%	1.35%	0.04	1.42%
	0.55%					
	¥5,492	@100万円				

- TTMにどれだけ上乗せされているか。
- 表下は、100万円分使ったときの差額。
- 信頼係数95%。

- 換算日が分かるもの（三井住友カード、JCBカード、セゾンカード）と換算日を合理的に推測できるものだけを集計。利用日を単純に換算日と仮定したものは集計に一切含めなかった。換算日の推測は、VISA/Master等のレートに各カード所定の公式手数料を乗せたレートと、カード明細記載のレートとの乖離が0または十分小さい日を換算日と推測した。十分小さい日がなく換算日を合理的に特定できなければ集計から除外。上記観測個数は、実際に最終の計算に使った取引のみの件数。除外取引を含まない。
- TTMは原則として三井住友カードなら三井住友銀行、DCカードなら三菱東京UFJ銀行を使った。ただし、三井住友銀行は台湾のTTMを公開していないなどの場合は適宜、日本の他行を使った。VISA/Master等のレートはそれ自体がTTMより悪いので、基準はあくまでもTTMとした。
- DCカードのMasterは持っているがVISAは持っていない。DCカードは手数料が上がってからは海外のショッピングで使うのをほぼ止めた。（キャッシングにはDCカードを専ら使っている。） Cf. [外貨でのショッピングご利用代金を円貨へ換算するための事務処理手数料改定のお知らせ（Visa・Mastercardをお持ちのお客さまへ）](#) | [クレジットカードなら三菱UFJニコス](#)

- もっとも、最近のデータを見ていて気付いたのだが、DCカードは公式の手数料率より実際の料率はかなり低いようだ。しかし、為替リスクが大きい傾向もある（実際の換算日（推測）が利用日からかなり離れている）ので痛し痒しだ。一概に有利なカードとは言えないようだ。

- 実際は発行会社その他の要因も影響しているようだが、分析するにはサンプルの多様性が全然足りない。
- この表は原本からリアルタイム更新なのでたまに実験中の異常値が表示されることもある。

- 三井住友カード、JCBカード、セゾンカードは換算日（三井住友カードが何日のレートを使ったか）が明細に表示されるのでデータを分析する上で重宝している。換算日が営業日でない場合、実際に適用されているのは翌営業日のレートみたいだ。（私個人のデータしかないなのでサンプルサイズが圧倒的に不足するのだが。）MasterやVISAのウェブでは営業日以外の換算レートも見られるけど、そのレートはTTMより上乗せがあるので、カードの実質手数料を比較する上でよい基準とは言えない。
- 換算日を書いてないカードでも推測は立つ。VISAまたはMasterのレートから実質手数料を計算して1.67%(あるいは各カードによる公式値)に一番近い日が換算日だろう。

- 各社公式情報

- 海外でのご利用範囲 | セゾンカードご利用ガイド | クレジットカードは永久不滅ポイントのセゾンカード
- 換算レートや海外事務手数料について教えてください。 | 楽天カード：よくあるご質問
- ご利用明細のご説明 | クレジットカードの三井住友VISAカード
- 外貨でのショッピングご利用代金を円貨へ換算するための事務処理手数料改定のお知らせ (Visa・Mastercardをお持ちのお客さまへ) | クレジットカードなら三菱UFJニコス

同日同店で競争

もう一つ、比較方法がある。同じ時刻に同じ店で複数のカードを使って相対評価する方式を試した。米国amazonでギフトカードをチャージして試している。これなら観測個数を増やすのが簡単だ。

	相対			TTM比(注)		
ANA Master (三井住友)	0.00%	±0.00	25	1.54%	±	
JAL DC Master	-0.17%	±0.27	22	1.39%	±	
ANA VISA Suica (三井住友)	0.29%	±0.10	24	1.86%	±	
ANA PASMO JCB (JCB)	0.08%	±0.21	18	1.60%	±	
Diners	0.89%	±2.08	3	2.15%	±	

右はTTMとの比較（いわば絶対評価。換算日基準）。左はANA Masterとの相対評価。各カードの請求書記載の「換算レート」（これは必ず載っている）を比較している。

- 見ての通り、JAL DC Masterの誤差が非常に大きく、統計的に意味がない。（その結果、JAL DC Masterを加えて「Master, VISA, JCB」という比較をすると統計的に意味がなくなってしまふ。）JAL DC Masterの誤差が大きいの、換算日（これは非公開）が利用日から（大きめに）離れているか、（そもそも）一定しないか、その両方か、だと推測する。ANA Masterとブランドは同じであるから、換算日が同じであればレートも同じになるはずだが、この通り全然同じにならない。

とりあえず、ANA MasterとANA VISA Suicaの比較では、VISAの方が手数料が高いと言える。どちらも三井住友カードで、換算日が明記されているので、TTM比で比べても無理がないカードだ。JCBの数字は誤差がまだ大きめであり、何とも言えない。

- TTMと比べる方式には課題がある。いつのTTMと比べるかだ。「換算日」は一つの基準になるが「換算日」が非公開のカードの方が多いただろう。また換算日のTTMが休業日で取得できないことも少なくない。そもそも、比較する日が利用日から離れば離れるほど、為替変動のリスクが大きくなり、ブランドやカード会社の手数料なのか、それとも為替変動による偶然なのか、区別が難しくなりそう。では利用日のTTMで比べたらどうか。それなら為替変動リスクを排除できそう。もちろん、利用日のTTMが手に入らないこともあるし、どの銀行のTTMを使うか、その他の課題はある。しかしそれは決めの問題であり致命的ではない。さて、実際に計算してみると、「そうだ」の通りには行かないようだ。
- 3つの方法には以下の差があるようだ。

換算日のTTMで比較する（絶対評価）	カード発行会社の上乗せ分を見る。
利用日のTTMで比較する（絶対評価）	カード発行会社とブランドの上乗せ分を見る。しかし誤差が大きく（為替変動や料率の変動や、その他未知の要因など全ての変動が一緒くたになっているためだろう）、カード毎の差も小さいため、解釈が難しい。
相対評価	為替変動リスクを含む消費者の負担を見る。誤差は為替変動リスクを見るのに適する。

- 相対比較の誤差の大小は、主に為替変動リスクを示している。利用日と換算日（公開非公開問わず）との間の日数が主に左右しているはずだ。
 - しかしその他に、換算レートが為替変動のトラッキング以外の要因に左右されている要素もあるだろう。根拠は、TTM比の誤差のANA PASMO JCB（換算日が明記されているカード）の値が特に小さいことだ。JCBの自社発行カードであり、VISAやMasterブランドのような中抜きする人が、他にいない（JCB単独のルールで換算レート=上乗せ率が決まっている）ことを示しているという推測が立つ。
- TTM比の誤差が小さいことは、カード会社が上乗せする率のルールに、消費者にとって未知の要因がないことを意味する。特に換算日を明記しているカードに妥当する。この点で、ANA PASMO JCB（換算日を明記）が

特に小さいのは、JCBの自社発行カードであり、VISAやMasterブランドのような中抜きする人が、他にいないことを示しているのかもしれない。

- ちなみに、実験開始当初は間隔を空けずに連続的に複数のカードを使ったが、8順目でカードブランド (Master) から承認拒否された。拒否したのはamazonや日本のカード会社ではない。カード会社に自分から電話して照会したら、カード会社側にはデータが来ておらず、従って拒否もしていないし、何も情報がない、とのことだった。Masterのカード (発行会社は違う) だけ拒否されたので、Master側 (Master段階というべきか) の拒否判定だと推測できる。Masterが拒否するとカード発行会社にも情報が (少なくとも利用日前後の時点では) 行かないようだ。 (一定期間でまとめて報告されることはあるかも。)
 - やり過ぎるとamazonからbanされるかも。 [アマゾンがアカウントを閉鎖する客とは - WSJ](#)
- JAL DC Masterの分散を人為的に小さくすることもできそうな感じ。利用日と換算日 (早ければ翌日) が、日本と米国いずれの休日にもかからないように利用すればよい。しかし、実際の買い物行動とは離れてしまう。ここではそういう操作はしていない。

TTM

TTMの入手先:

- [外国為替情報 : 三井住友銀行](#)
- [三菱UFJリサーチ&コンサルティング | 外国為替相場 | 1990年以降の為替相場](#)
- TTM (仲値) というわけではないが、 [Pacific Exchange Rate Service - Database Retrieval System](#)

自動化

自動化しないと、誤記する。

- 三井住友銀行

```
#!/usr/bin/perl
use utf8;
use strict;
use warnings;

my $url = 'http://www.smbc.co.jp/market/backnumber/fixing/';

# 今月
my $thisMonth = '//*[@id="tabbox1"]/div//li/a/@href';
# 前月
my $lastMonth = '//*[@id="tabbox2"]/div//li/a/@href';
use URI;
my $u = URI->new($url);
use HTML::TreeBuilder::XPath;
my $tree = HTML::TreeBuilder->new_from_url($url);
my @href = $tree->findvalues($lastMonth);
push(@href, $tree->findvalues($thisMonth));
use LWP::UserAgent ();
my $ua = LWP::UserAgent->new;

for my $pdfpath (@href) {
    my $uri = sprintf qq(http://%s%s), $u->host, $pdfpath;
    my $response = $ua->get($uri);
    if ($response->is_success) {
        my $pdfbody = $response->decoded_content;
        my $tmpfile = "/tmp/msbcrate.$$";
        open(my $fh, '>', "$tmpfile") or die "$tmpfile:#!";
        print $fh $pdfbody;
        close $fh;
        open($fh, "pdftotext -layout $tmpfile - |") or die "pipe:#!";
        my @content = <$fh>;
        close $fh;
        unlink $tmpfile;
        if (my ($date, %rate) = &msbcrate_pdf(@content)) {
            my $ttm = ($rate['usd']->{'tts'} + $rate['usd']->{'ttb'})/2;
            printf qq(%s¥t%.2f¥n), $date, $ttm;
            #my $insert =
            #"INSERT IGNORE INTO usd VALUES ('$date', 'SM', '$ttm')";
            #$db->do($insert) or die $db->errstr;
        } else {
            print STDERR "failed in $pdfpath¥n";
        }
    }
}
```

```
}
# $db->disconnect;

exit;

sub msbcrate_pdf {
    my @lines = @_;
    my $date = '';
    my %rate = ();
    my %month = qw(
Jan   1
Feb   2
Mar   3
Apr   4
May   5
Jun   6
Jul   7
Aug   8
Sept  9
Sep   9
Oct  10
Nov  11
Dec  12
);

    for (@lines) {
        # Date:   May. 25, 2018
        if (my @temp = (/Date:¥s+([a-zA-Z]+)¥D+(¥d+)¥D+(¥d+)/)) {
            my $month = $month{$temp[0]} // 'fail';
            $date = "$temp[2]/$month/$temp[1]";
            next;
        }
        if (/U¥. S¥. A¥. ¥s+USD¥s+1¥s+([¥d¥. ]+)¥s+([¥d¥. ]+)¥s+([¥d¥. ]+)¥s+([¥d¥. ]+)¥s+([¥d¥. ]+)¥s+([¥d¥. ]+)¥s+([¥d¥. ]+)/) {
            $rate{'usd'}->{'tts'} = $1;
            $rate{'usd'}->{'ttb'} = $3;
            next;
        }
    }
    if ($date eq '' || !defined $rate{'usd'}->{'tts'}) {
        return;
    } else {
        return $date, %rate;
    }
}
}
```

- 三菱UFJリサーチ&コンサルティング

```
#!/usr/bin/perl
use utf8;
use strict;
use warnings;

use Time::Piece;
use Time::Seconds;
my $t = localtime;
my $days = 20; # 過去〇日分取得する。

use LWP::UserAgent ();
use HTML::TreeBuilder::XPath;
my $baseurl = 'http://www.murc-kawasesouba.jp/fx/past/index.php';
# http://www.murc-kawasesouba.jp/fx/past/index.php?id=180419

for (1..$days) {
    my $date = sprintf qq(%02d%02d%02d), $t->yy, $t->mon, $t->mday;
    my $url = $baseurl . '?id=' . $date;
    my $ua = LWP::UserAgent->new;
    # 営業日以外は「本日の為替相場」に転送される。
    # http://www.murc-kawasesouba.jp/fx/index.php
    $ua->max_redirect(0);
    my $response = $ua->get($url);
    my $root = HTML::TreeBuilder->new();
    if ($response->is_success) {
        $root->parse_content($response->decoded_content);
        my $tts = $root->findvalue('//*[@id="main"]/table[1]/tr[2]/td[4]');
        my $ttb = $root->findvalue('//*[@id="main"]/table[1]/tr[2]/td[5]');
        my $ttm = ($tts+$ttb)/2;
        my $date = $t->ydm("/");
    }
}
```

```

        printf qq(%s¥t%.2f¥n), $date, $ttm;
        #my $insert =
        #"INSERT IGNORE INTO usd VALUES ('$date', 'MU', '$ttm')";
        # $db->do($insert) or die $db->errstr;
    }
    $t -= ONE_DAY;
}
# $db->disconnect;
exit;

```

カード明細

速報はCSVでダウンロードできないカードが意外とある。そういうのは[勝手にボタンを付ける](#)。Amexは単純なスクレイプでは取得できない（ページがJavaScriptで動的に作られる）ので、headless Chrome + Puppeteer (Node.js)で取得している。

Amex自動化

```

const puppeteer = require('puppeteer');
const fs = require('fs');

(async() => {
    const browser = await puppeteer.launch({
        //headless: false,
        args: [
            '--no-sandbox',
        ]
    });
    const page = await browser.newPage();
    await page.setViewport({width: 1024, height: 768});

    await page.goto('https://global.americanexpress.com/myca/logon/japa/action/LogonHandler?
request_type=LogonHandler&Face=ja_JP&inav=jp_utility_login', { waitUntil: 'networkidle2' });

    await page.waitForSelector('input#lilo_userName', {waitUntil: 'networkidle2', timeout: 10000});
    await page.focus("input#lilo_userName");
    await page.type("input#lilo_userName", 'your ID');
    await page.focus("input#lilo_password");
    await page.type("input#lilo_password", 'your passwd');
    await page.click('#lilo_formSubmit');

    await page.waitForNavigation({waitUntil: 'load'});
    await page.waitForSelector('#balance-summary-table', {waitUntil: 'networkidle2', timeout: 2500});
    // カードによって違うので要修正
    await page.click('#balance-summary-table tr.recent-debits > td.description > a');

    await page.waitForNavigation({waitUntil: 'load'});
    await page.waitForSelector('#transaction-table', {waitUntil: 'networkidle2', timeout: 7000});

    //tr要素のidを取得。transaction id
    let tid = await page.$$eval('#transaction-table tr[id]', list => {
        return list.map(date => date.id);
    });
    let lines = [];

    // transaction id毎に
    // 各決済のtrは2段からなる。
    // クリックによる展開前と展開後の詳細
    // idが違う。
    for (let i in tid) {
        await page.click("#transaction-table tr#" + tid[i] + " td.date");
        await page.waitForSelector("#transaction-table tr#" + tid[i] + " tr.p.amount.ng-binding", {waitUntil:
'networkidle2', timeout: 7000});
        // 詳細欄のtransaction id取得
        // 例: #etd-details-content-AT181580...
        let dom = '#transaction-table tr#' + tid[i] + ' tr[id]';
        let tidd = await page.$eval(dom, e => e.id);

        // dateと請求額は1段目のtr
        dom = 'tr#' + tid[i] + ' td.date span.ng-binding';
        let date = await page.$eval(dom, e => e.innerText);
        // #trans-... > td.date > div > span.ng-binding

```

```

    let shop = await page.$eval("tr#" + tidd + " p.header.ng-binding", e => e.innerText);
    // #etd-details-content-AT1815... > div > div.etdContent.ng-scope > div.etdBlock.ng-scope > p.header.ng-
binding

    let local = await page.$eval("tr#" + tidd + " p.amount > span:nth-child(1)", e => e.innerText);
    // #etd-details-content-AT1815... > div > div.etdDetails.ng-scope > div > div:nth-child(2) > p.amount >
span:nth-child(1)

    let curr = await page.$eval("tr#" + tidd + " p.amount > span:nth-child(2)", e => e.innerText);
    // #etd-details-content-AT181... > div > div.etdDetails.ng-scope > div > div:nth-child(2) > p.amount >
span:nth-child(2)

    dom = 'tr#' + tid[i] + ' td.amount > div > span';
    let jpy = await page.$eval(dom, e => e.innerText);
    // #trans-AT18158... > td.amount > div > span

    let rate_date = await page.$eval("tr#" + tidd + " div.etdContent.ng-scope > div:nth-child(2) > div >
p.ng-binding.ng-scope", e => e.innerText);
    // #etd-details-content-AT18156... > div > div.etdContent.ng-scope > div:nth-child(2) > div > p.ng-
binding.ng-scope

    let rate = await page.$eval("tr#" + tidd + " p.amount.ng-binding", e => e.innerText);
    // #etd-details-content-AT181... > div > div.etdDetails.ng-scope > div > div:nth-child(3) >
p.amount.ng-binding

    let line = [date, shop, local, curr, jpy, rate_date, rate].join(', ');
    console.log(line);
    lines.push(line);
  }
  fs.writeFile('amex.csv', lines.join("\n"), (err) => {
    if (err) throw err;
  });
  //ログアウト
  await page.click('#jp_utility_login');
  await browser.close();
})();

```



Amexの意外な競争力


Amexの決済インフラは他社に比べ圧倒的に強いかもしれない。Amexの自社発行カード（日本法人）は、試した範囲では、他のイシューア（発行会社。三井住友カードや楽天カード、DCカードなど）やブランド（VISA、Master、JCBなど）に比べ、カードを利用してから明細に出るまでの時間が、圧倒的に速い。（今回、明細の確認はロボットで自動化しているので、傾向を統計的に把握できる。）クレジットカードの仕組みはもう何十年も前に作られたもので、本来リアルタイム決済を想定したものでは全くない。時代の趨勢に答えてリアルタイムで処理するには、多大なシステム投資が必要だ。しかも、自社だけの投資で終わらず、販売店との連携も必要になる。また、クロスボーダーの決済はどんどん増えるだろう。Amexでは、日本発行カードでも日本の営業時間に制約されないことも見て取れる。

Amexの他では、三井住友カード（VISA、Master）が結構健闘していた。それでも、日曜はデータが動かないことが見て取れるなど、システム設計の古さを推知できる。日本時間の午前中か否かで処理のバッチが切り替わる（Master）ことも推知できる。VISAかMasterかでバッチの挙動も異なる。自社ブランドでないので自分でコントロール出来ないのだろう。今回のカードでは、JCBは自社ブランド・自社発行のカードだが、速度に特に目立った特徴はない。三井住友トラストクラブ株式会社（三井住友信託銀行の100%子会社。旧、シティカードジャパン株式会社）のDinersは、今回、システム更新後のみで試しているが、かなり遅い。

こうしたことも、個人で調べるのは時間も金もかかるが、マネーフォワードなら簡単に分析できる。研究者には垂涎の的だろう。



このページのアーカイブ

- [このページの2018年 5月12日時点のPDFのハッシュ値\(md5\)](#)
4436db3ae118bf23f5e77b3b53456b2bdeb25b1e at [Twitter](#)
- ちなみに、ハッシュ値はWindows標準のコマンドで求めることができる。

Last-modified: 2018-05-30 (水) 23:04:41 (12d)

Site admin: admin@6pb.info

PukiWiki 1.5.1 © 2001-2016 [PukiWiki Development Team](#). Powered by PHP 7.0.30. HTML convert time: 0.048 sec.