

# Sunpath Dashboard Application Manual

StatusQuota

September 19, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Access Information . . . . .	3
1.2	Plotly/Dash . . . . .	4
1.3	Navigation . . . . .	4
1.4	Data Collection and Information . . . . .	4
1.5	Plot/Graph Visuals and Features . . . . .	5
1.6	Table Visuals and Features . . . . .	5
<b>2</b>	<b>Admin Bank Reconciliation</b>	<b>6</b>
2.1	Summary . . . . .	6
2.1.1	How to Use . . . . .	6
2.1.2	Value Definitions . . . . .	6
2.2	Details . . . . .	7
2.2.1	How to Use . . . . .	7
2.2.2	Value Definitions . . . . .	8
<b>3</b>	<b>Funder Bank Reconciliation</b>	<b>8</b>
3.1	Summary . . . . .	8
3.1.1	How to Use . . . . .	8
3.2	Value Definition . . . . .	9
3.3	Details . . . . .	9
3.3.1	How to Use . . . . .	9
3.3.2	Value Definitions . . . . .	10
<b>4</b>	<b>Scenario Modeling</b>	<b>11</b>

4.1	Summary . . . . .	11
4.1.1	How to Use . . . . .	11
4.1.2	Visuals and Plots . . . . .	12
4.1.3	Value Definitions and Calculations . . . . .	13
4.2	Admin and Vendor Details . . . . .	14
4.2.1	How to Use . . . . .	14
4.2.2	Values Definitions and Calculations . . . . .	16
<b>A</b>	<b>SQL Queries</b>	<b>20</b>
A.1	Bank Reconciliation Admin Details . . . . .	20
A.2	Bank Reconciliation Funder Summary and Details . . . . .	20
A.3	Scenario Modeling Summary and Details . . . . .	20

# 1 Introduction

This manual explains the usage and functionality of the application. For each dashboard component the documentation will include: an overview description and functionality, screenshots of usage, and underlying calculations for values shown, if applicable. Any duplicate calculations or references will be referenced back to the first mention.

The manual is structured by sections. First, the introduction gives a basic explanation on the overall usage within the application. It is structured as such: access information, application foundation, navigation, data collection, plot features, and table features. Next, the admin and funder bank reconciliation dashboards will be explained with screenshots, descriptions, and how/where the values are being extracted. The scenario modeling dashboard will be explained with screenshots and descriptions for the various admin and vendor details. Finally, an appendix includes any extra details not required for functionality, but it gives more detailed info such as sql queries or python code.

Also, it is important to note that each time you navigate away from a dashboard the dashboard **”resets”** and goes back to original state it was before the user navigated to it.

## 1.1 Access Information

In order to access the application the user must navigate to the link listed in the following credentials. The authentication uses Google OAuth technology, and as a result it will ask you to authenticate with a google account email. Once you authenticate, the authentication will be stored into the cookies, and as a result you will no longer be asked to authenticate on the machine until you clear said cookies. The only restriction is that the emails that can be allowed to access the application must be manually inputted into a file on the server.

The user credentials needed for the test versions for the application are listed below:

1. webapp temp address (will change for final version):  
<http://alexpan.ktechboston.com/>
2. dummy user email: `spfadmnr.test@gmail.com`
3. password: `webappP@55`

## 1.2 Plotly/Dash

The application foundation uses the python framework known as Dash. Dash uses the Plotly framework for visualizations. It is a web-based interface that is used for building interactive analytical web applications. It uses various UI elements like dropdowns, sliders, and graphs in combination with Python code. This allows for a very powerful tool that can combine web-based applications with coding elements. As a result, it provides a foundation for interactive visualizations in a medium that can be simple, easy to use and learn.

## 1.3 Navigation

The "Home Page" will be the main portal/location to navigate the various dashboard components. Each dashboard component has a link in the top left corner that will directly take the user back to the home page. When a user wants to navigate to a dashboard, they must click on the link to the desired dashboard component at the home page. **It is important that one must select the "Home Page" link to navigate back to the home portal in order to load the page. This is to insure correct loading of application elements.**

## 1.4 Data Collection and Information

The data for the application is gathered directly from csv files that are generated by scripts. The scripts generate csv files from querying commands from the Sunpath database **SPAdmin** and **FundingPaylink\_Live**. As long as the Sunpath database is updated, so will the data that is shown in the dashboards. Here is a list of databases, tables, and the corresponding dashboards associated with the table:

Database: **SPAdmin**

1. SPA\_SPBankingStat (Bank Reconciliation)
2. SPA\_SPDepositsStat (Bank Reconciliation)
3. SPA\_Funded\_Contracts (Bank Reconciliation)
4. SPA\_SPPlugStat (Bank Reconciliation)
5. SPA\_SPPaymentsStat (Bank Reconciliation)

Database: **FundingPaylink\_Live**

1. SPF\_Funding\_Transaction\_Log (Bank Reconciliation, Scenario Modeling)
2. SPF\_Funding\_Transaction\_Codes (Bank Reconciliation, Scenario Modeling)
3. Daily\_Extract(Scenario Modeling)
4. Admin\_Funding (Scenario Modeling)
5. Seller\_Funding (Scenario Modeling)
6. StoneEagle\_All\_Customer\_Info (Scenario Modeling)
7. Seller\_Info\_Funding\_Approved\_Partners (Scenario Modeling)
8. Seller\_Info\_Funding\_Parameters (Scenario Modeling)

## 1.5 Plot/Graph Visuals and Features

The visualizations and plots generated from the application and dashboards can be interacted with by the user. All plots and visuals have the following attributes: hover tool, functionality features (zoom, pan, box select, axes reset and autoscale), download to png, ability to edit in plotly's chart studio, and much more. A convenient toolbar is visible once the user hovers on the visual.

## 1.6 Table Visuals and Features

Most tables in the application also come with some basic customizable features. A user can take a column header and resize to their preferred width. Also, a user can sort a table by selecting a column header and choose to sort by ascending or descending values. And unless specified, some tables allow the user to edit the table by choosing their own input values.

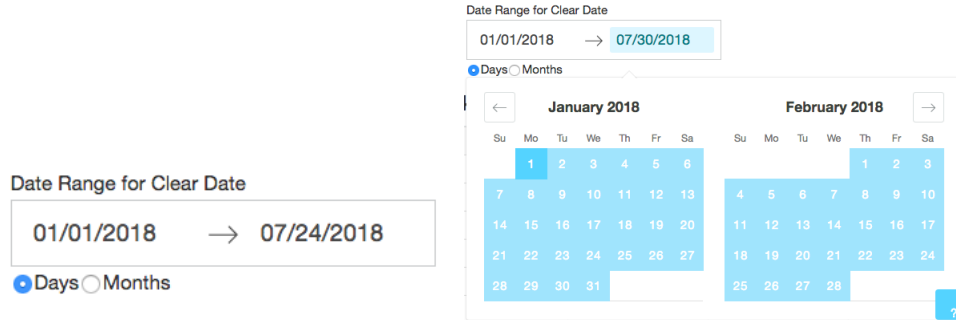
## 2 Admin Bank Reconciliation

### 2.1 Summary

This dashboard component allows the user to see the admin version of the bank reconciliation summary dashboard. It shows a table and column names for "Bank of America" and "Sunpath" tables. The "Bank of America" columns are the various descriptions for account types: *Days/Months*, *Ops*, *Claims*, *INS*, *Payroll*, *Escrow*, and *Total*. The "Sunpath" columns are the types/database tables : *Days/Months*, *Deposits*, *Insurance*, *Plugs*, *Packs Payments*, and *Total*. The *Variance* column value is the difference between the "Bank of America:Total" and "Sunpath:Total".

#### 2.1.1 How to Use

1. Edit the Date Range for the "Clear Date" by selecting the dates in the dropdown calendar. One can also modify the view to show "Days" or "Months"



2. Once a date range is selected, the table will fill with necessary information, which can be scrollable

#### 2.1.2 Value Definitions

For the Bank of America table, the column values: *Ops*, *Claims*, *INS*, *Payroll*, and *Escrow* are *Account* types within the **SPA.SPDepositsStat** database table. The Bank of America's rows are organized by days or months starting from the first *ClearDate* in the selected date range. The Bank of America account's values are the sums from the *Amount* column in the database table on the selected day or month. The Bank of America column *Total* is defined as

$$Total = Ops + Claims + INS + Payroll + Escrow \quad (1)$$

Similarly, the Sunpath table is organized by day or month. The Sunpath column values: *Deposits*, *Insurance*, *Plugs*, and *PacksPayments* are taken from the database tables **SPA\_SPDepositsStat**, **SPA\_Funded\_Contracts**, **SPA\_SPPlugStat**, and **SPA\_SPPaymentsStat**. The values are sums of either the *PaymentAmount* or *Amount* columns in the database tables except for *Insurance*, where the value is taken from *InsuranceReserveAmount* in the database table. The Sunpath total is defined as

$$Total = Deposits + Insurance + Plugs + PacksPayments \quad (2)$$

The final value *Variance* is defined as the difference between these two totals

$$Variance = \text{equation 1} - \text{equation 2} \quad (3)$$

## 2.2 Details

This dashboard component shows the detail summary of the various tables/account types/tables: Plugs, Deposits, Payments, Insurance, Banking Transaction. The details help provide basic summaries and include further notes, descriptions, payee, vendor, and/or group name associated with the specific amount in the database entry. In order to provide this more detail summary, the date range can only be organized by *ClearDate* in units of days. Due to the size of the tables, all tables are displayed and can be scrolled through.

### 2.2.1 How to Use

1. Edit the Date Range for the "Clear Date" from the dropdown calendar.

The screenshot displays the 'Date Range for Clear Date' interface. At the top, a date range is selected from 01/01/2018 to 07/30/2018. Below this, a calendar view for January and February 2018 is shown. The calendar has columns for days of the week (Su, Mo, Tu, We, Th, Fr, Sa) and rows for dates. The dates 1 through 31 are visible, with the 1st of January highlighted in blue. A question mark icon is visible in the bottom right corner of the calendar view.

2. Information will be filled in the tables once a date range is selected.



### 2.2.2 Value Definitions

All tables and values were generated using the sql queries found in appendix A.1 at the end of the document. Some values were renamed, such as disbursetype to groupname, PaidTo to Payee, etc. This can all be seen in the appendix.

## 3 Funder Bank Reconciliation

### 3.1 Summary

This dashboard component shows info intended for the funder side of the business. It functions and operates similarly to the admin component, but differs in the information shown. It shows the user the totals for the various types/accounts for each day or month and shows the variance between the totals from the accounts and the bank amount total from the transaction logs. The transaction logs were organized using the sql query in appendix A.2. The data collected is from the database tables **SPF\_Funding\_Transaction\_Log** and **SPA\_FundingBankStat**.

#### 3.1.1 How to Use

1. Edit the Date Range for the "Clear Date" from the dropdown calendar. One can also modify the view to show "Days" or "Months"

The screenshot displays the 'Date Range for Clear Date' interface. At the top, there is a date range selector showing '01/01/2018' to '07/30/2018'. Below this, there are radio buttons for 'Days' (selected) and 'Months'. The main part of the interface shows a calendar view for January and February 2018. The calendar for January 2018 shows dates from 1 to 31, with the 1st highlighted in blue. The calendar for February 2018 shows dates from 1 to 28, with the 1st highlighted in blue. A blue question mark icon is visible in the bottom right corner of the calendar area.

2. Once a date range is selected, the table will fill with necessary information, which can be scrollable

### 3.2 Value Definition

As mentioned, the data collected is from the database tables **SPF\_Funding\_Transaction\_Log** and **SPA\_FundingBankStat**. The *BankAmtTotal* is collected from the **SPA\_FundingBankStat**. The second table's values are compiled through the transaction log's *TxDescription* and *TxAmount*. The *Total* for each day is the addition of all the values, defined as:

$$\begin{aligned} Total = & DepositsAmt + InsuranceAmt + PaymentAmt \\ & + PlugAmt + CustCollAmt + CollectAmt \end{aligned} \quad (4)$$

The *Variance* is the difference between the *BankAmtTotal* and *Total*:

$$Variance = BankAmtTotal - Total \quad (5)$$

### 3.3 Details

Similar but slightly different to the admin bank reconciliation section details 2.2, it includes accounts/types from customer collections and collections and the details were gathered from different tables. The tables shows the *Amount* grouped by the *ClearDate*, *Description*, and/or *Vendor*

#### 3.3.1 How to Use

1. Edit the Date Range for the "Clear Date"

Date Range for Clear Date

01/01/2018 → 07/30/2018

←
**January 2018**
February 2018
→

Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6					1	2	3
7	8	9	10	11	12	13	4	5	6	7	8	9	10
14	15	16	17	18	19	20	11	12	13	14	15	16	17
21	22	23	24	25	26	27	18	19	20	21	22	23	24
28	29	30	31				25	26	27	28			?

Date Range for Clear Date

01/01/2018 → 07/24/2018

2. Information will be filled in the tables once a date range is selected.

### 3.3.2 Value Definitions

In order to separate the descriptions into their respective account types within the database table **SPA\_FundingBankStat**. A key was created to assist in this process. Currently, this key is static. Any new descriptions would require manual updating/editing. The description key is defined as

```
DESCR_KEYS =
Administrator Funding : Deposits,
Insurance Reserve Funding : Insurance,
Paylink Chargeback Fee : Customer Collections,
Paylink Customer Collection : Customer Collections,
Paylink Customer Collection Reversal : Customer Collections,
Paylink Processing Fee : Customer Collections,
Paylink Processing Fee Reversal : Customer Collections,
Paylink Returned Payment Fee : Customer Collections,
RC (Debit Seller Wire) : Deposits,
Returned Premium - Admin : Deposits,
Returned Premium - Ins Reserve : Deposits,
Reverse Administrator Funding : Payments,
Reverse Insurance Reserve Funding : Collections,
Reverse Returned Premium - Admin : Collections,
Reverse Returned Premium - Ins Reserve : Collections,
Reverse Seller Funding : Collections,
Seller Funding : Payments
```

## 4 Scenario Modeling

The scenario modeling dashboards consist of three components: summary, admin details, and vendor details.

### 4.1 Summary

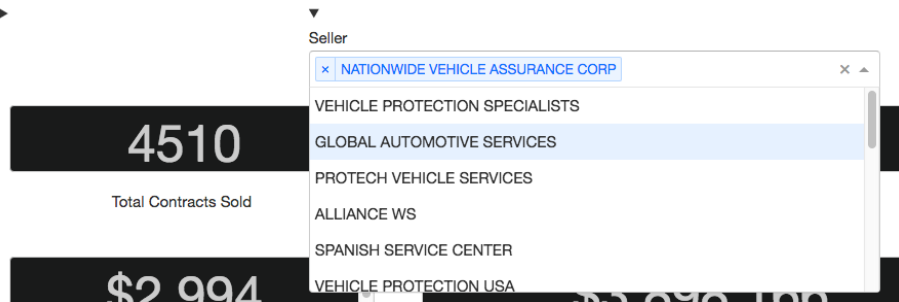
This dashboard component consists of values and visuals to give the user a basic summary of a vendor, funder cohorts. The values shown are "Total Contracts Sold", "Total Face Value Sold", "Avg. Contracts Sold, Month", "Avg. Face Value", "Total Seller Advance Received", "Avg. Contracts Sold, 3 Months", and "Growth Rate, M-O-M (Recent ,Complete)." The two interactive visuals show the Cohort's Month Sales Volume and the Cohort's Current Cancel Rates per Payments Made.

#### 4.1.1 How to Use

1. Click on the arrow to toggle the funder field from visible and not visible. In the field, the user can select 1 or more funders from the dropdown list



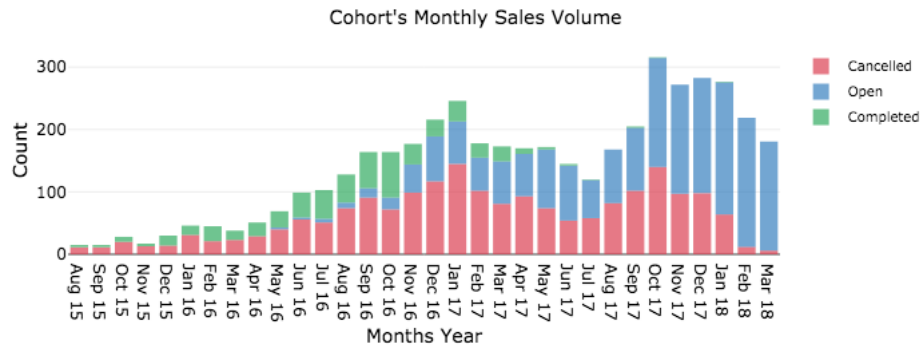
2. Click on the second arrow and toggle the vendor field from visible and not visible. In the field, the user can select 1 or more funders from the dropdown list



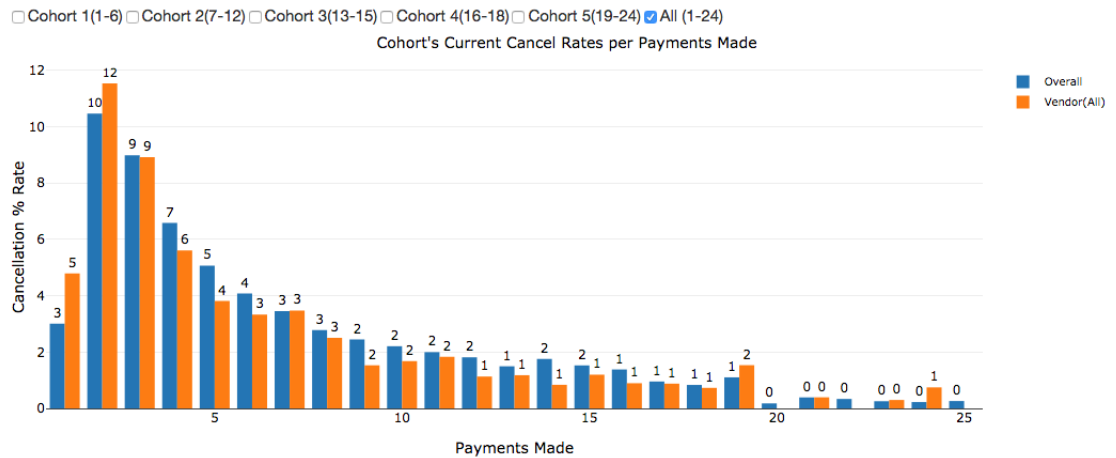
3. The fields will fill with numbers and the visuals will display

#### 4.1.2 Visuals and Plots

1. The cohort's monthly sales volume shows the count vs. month and year. The plot separates counts by cancelled, open, and completed contracts



2. the cancel rates per payments made shows the selected vendor cancel rates compared with all the vendors. Also the user can select 1 or more cohort(installment term lengths) to look at specific contract terms and displayed above each bar/bin is the percent(rounded).



### 4.1.3 Value Definitions and Calculations

The cancel curve plot visual is created by using rules from the cancelillustration.pdf; a brief example is shown below. The numerator is defined as "Number

	Number Cancelled Per Payment		Qualified Funded Contracts*		Cancellation Experience Per Payment
1 »	21	/	160	=	13.125%
2 »	12	/	150	=	8.667%
3 »	8	/	140	=	5.714%
4 »	6	/	130	=	4.615%
5 »	5	/	120	=	4.167%
6 »	3	/	110	=	2.727%
7 »	2	/	100	=	2.000%
8 »	1	/	90	=	1.111%
9 »	0	/	80	=	0.000%
10 »	1	/	70	=	1.428%
11 »	1	/	60	=	1.667%

Cancelled Per Payment" and the denominator is defined as "Qualified Funded Contracts", which is defined below in a small figure. The resulting fraction is

\*Qualified Funded Contracts  
 For VSCs with 1 payment made, qualified contracts are those with an effective date older than today's date MINUS 4 months.  
 For VSCs with 2 payments made, qualified contracts are those with an effective date older than today's date MINUS 5 months.  
 For VSCs with 3 payments made, qualified contracts are those with an effective date older than today's date MINUS 6 months.  
 For VSCs with 4 payments made, qualified contracts are those with an effective date older than today's date MINUS 7 months.  
 ...  
 For VSCs with 23 payments made, qualified contracts are those with an effective date older than today's date MINUS 26 months.

the "Cancellation Experience Per Payment."

The monthly sales volume histogram's bins for a cohort are calculated by grouping by month, year and contract status: completed, cancelled, or open. Finally, the values shown in fields are calculated and defined as:

1. Total Contracts Sold = Total # contracts in a cohort's historical data
2. Total Face Value Sold = Total amount financed from all contracts in a cohort's historical data
3. Avg. Contracts, Month = The mean or average number of contracts sold per month based on a cohort's historical data
4. Avg. Face Value = The mean or average amount financed for a contract based on a cohort's historical data

5. Total Seller Advance = Total Seller Advance Amount from each contract based on a cohort's historical data
6. Avg. Contracts Sold, 3 last months = The mean or average number of contracts sold in the most recent 3 months of a cohort
7. Growth Rate = Calculates the rate of contracts sold from 2 months from the most recent month. Example. if now = April 2018, we look at the growth rate from January 2018 to February 2018. This was chosen because 2 months from the most recent month allowed the database to be properly updated with data.

## 4.2 Admin and Vendor Details

This dashboard component contains the tables and values for admin or vendor details. The admin dashboard component includes "Loss Ratio" and "Expected IRR" columns in the tables, but the vendor component does not include these. Also, a general rule is applied for sellers that do not have many contracts under their historical performance. If a seller or cohort has less than 75 contracts then we use a "Sunpath Average", which uses the data from taking all the contracts in Sunpath's historical database.

**Note 1:** Some of the fields will not load on first time data is calculated, depending on the data queried. This can easily be solved by reselecting the seller, funder, or editing an editable field and hitting "Enter" to reload and recalculate.

**Note 2:** For the purpose of this manual and section, a funder and seller combination will be defined as a group and the installment terms will be defined as a cohort.

### 4.2.1 How to Use

1. Click on the arrow to toggle the funder field from visible and not visible. In the field, the user can select 1 or more funders from the dropdown list
2. Click on the second arrow and toggle the vendor field from visible and not visible. In the field, the user can select 1 or more funders from the dropdown list
3. The user can modify the "Contract Fee" value by changing the field (Default is 50)
4. The fields will fill with numbers and the visuals will display, this may take a few seconds where the length it takes to load depends on the cohort's data

▼

Funder

Select a Fu..▼

▼

Seller

Select a Seller ▼

Contract Fee

50

- The leftmost columns: "Cancel Reserve %", "Discount Amount %", and "Contracts,Month" can be modified **Highlighted in Red**
- Changing these values will modify the columns corresponding to the related related table on the right **Highlighted in Blue**

Installment Terms	Cancel Reserve %	Discount Amt %	Net Amount,Contract	Expected IRR %
1	28.13	4.7	1042	
2-6	30.2	10.99	1174	7.32
7-12	34.85	11.04	601	-4.83
13-15	44.5	13.84	724	-1.4
16-18	49.73	16.18	557	-4.84
19-24	54.16	23.08	354	-9.16

Installment Terms	Cancel Reserve %	Discount Amt %	Contracts,Month	Net Amt,Contract	Accuring Net,Month	Expected IRR %
1	28.13	4.7	21	0	0	
2-6	30.2	10.99	14	0	0	
7-12	34.85	11.04	34	645	0	-4.8
13-15	44.5	13.84	9	739	1478	-1.99
16-18	49.73	16.18	79	534	1068	-5.31
19-24	54.16	23.08	39	1305	26100	-0.27



#### 4.2.2 Values Definitions and Calculations

This section gives the description and calculations for this dashboard's values. Basic summary of the values are defined below, where a cohort is defined as the installment term range:

1. Installment Term: Funder & Vendor groups divided by contract term lengths 1,2-6,7-12,13-15,16-18,19-24
2. Contract Sold: Number of contracts sold in the installment term cohort
3. % Contracts Sold: Percent of contracts sold
4. Cancel Reserve %: Cancel reserve amount percent
5. Discount Amount %: Discount amount percent
6. Net Amount: The Net Holdback for the installment term cohort
7. Loss Ratio:  $\frac{APR+EPR}{\text{Total Cancel Reserve Amount}}$
8. Net Amount, Contract: The net holdback per contract in installment term cohort
9. Current IRR %: The internal rate of return as a percentage for contracts in the cohort up to their current state (this is not a projection)
10. Cumulative Net Deficit/Surplus: The Cumulative Net Holdback amount for the funder, seller group
11. Cumulative Cancel Reserve: The total cancel reserve for the funder, seller group
12. Avg. Contracts Sold, Month: The average contracts sold per month for the funder, seller group
13. Contracts, Month: This editable field is the number of contracts sold per month to calculate the accruing net holdback per month
14. Net Amt, Contract: The net holdback amount on a per contract basis for the installment term cohort
15. Accruing Net, Month: The accruing net holdback per month based on the net holdback amount per contract

The calculations for many of these values will be shown below, as well as the variables used in the calculations:

1.  $TermDays$  = term of a contract in days

2.  $EffectiveDate$  = the Effective Date of a contract
3.  $DueDate$  = the due date of next payment for a contract
4.  $CancelRsv$  = cancel reserve of a contract
5.  $SellerCost$  = the seller cost of a contract
6.  $InstallAmt$  = installment amount for a contract
7.  $DiscountAmt$  = discount amount of a contract
8.  $PaymentsMade$  = the total payments paid by customer for a contract
9.  $TotalCancelRsvAmt$  = the total cancel reserve amount for cohort
10.  $ProbNC$  = probability of a contract not cancelling
11.  $ProbC$  = probability of cancelling
12.  $RP$  = returned premium (unweighted)
13.  $Term$  = a contract installment term length
14.  $fee$  = contract fee
15.  $AdminFundAmt$  = admin funding amount
16.  $InsRsvAmt$  = insurance reserve funding amount
17.  $APR = InstallAmt \times PaymentsMade$  for cohort
18.  $EPR$  = total expected projected receivable for cohort
19.  $DayUtilized$  = Cancel Date - Effective Date of a contract in days
20.  $VUR = \frac{DayUtilized}{TermDays}$
21.  $VendorRate$  = vendor specific rate
22.  $ProratedFee = VendorRate \times DiscountAmt$
23.  $PaymentPlanAmt = InstallAmt \times Term$
24.  $InstallAmtRec = InstallAmt \times PaymentsMade$ , amount received currently for a contract
25.  $AmtOwedSPF = (1 - VUR)(AdminFundAmt) - fee$
26.  $AmtOwedIns = (1 - VUR)(InsRsvAmt)$

The main calculation for *NetHoldback* is different, depending on if a contract cancels or completes:

$$NetHoldback = Total + ReturnedPremium \quad (6)$$

where *Total* and *ReturnedPremium* is defined as

$$\begin{aligned} Total = & CancelRsv - PaymentPlanAmt + AmtOwedSPF \\ & + AmtOwedIns + DiscountAmt - ProratedFee \\ & + InstallAmtRec \end{aligned} \quad (7)$$

$$Fraction(Days) = \left\lceil \frac{TermDays + [EffectiveDate - (DueDate + 30days)]}{TermDays} \right\rceil$$

$$ReturnedPremium = Fraction(Days) \times SellerCost - 50 \quad (8)$$

The *ExpectedValue* is the expected projected receivable for an contract at some payment  $i$  in its lifetime. A cancelled contract's expected value would be the returned premium that is taken from contract fields within the database. A completed contract would not expect any further payment. Finally, an open contract would expect to receive payments up to the last payment,  $j$ , and if it cancels it would have a returned premium, which is calculated from equation 8.

The expected projected receivable is a function of a contract's *InstallAmt*, *Term*, *PaymentsMade*, *EffectiveDate*, and *SellerCost* for returned premium calculation. The method and algorithm utilizes a Monte Carlo to create a simulation of contracts with installment term  $N = 1, 2, 3, 4, \dots, 24$ . It uses the contracts within the database to calculate the conditional probabilities for contracts with term  $N$  given installments paid  $i$ . It uses these probabilities to run a simulation for each individual contract term 100,000 times, and then uses the information to get an expected last payment  $j$  as a function of term  $N$  and installments paid  $i$ . The value used for last payment is one standard deviation above the mean, as this value gave us the "best" precision based on backtesting analysis between 50%-80% depending on the contract term, a contract's lifetime, and the amount of contracts in the backtesting cohort.

$$last\ payment = Probability(N, i) \quad (9)$$

The IRR calculation uses a built in function available in Python. This is not a projection as that would be beyond the dashboard's purpose, and a projected IRR would require more time and computation power that is not available with a real time calculation product. As a result, we can compute the expected IRR % as a function of cancel reserve and discount amount. In order to calculate this value, the contract's cashflows had to all begin at a similar point in time, which we call  $T_0$ . All the contracts in the cohort begin at  $T_0$  and cashflows are added together to create one cashflow. This final cashflow is then used to calculate the current IRR %.

The other value/column calculations are:

$$\% \text{ Contracts Sold} = \frac{\text{Contracts Sold in Cohort}}{\text{Total Number of Contracts in Group}} \quad (10)$$

$$\text{LossRatio \%} = \frac{APR + EPR}{TotalCancelRsvAmt} \quad (11)$$

$$\text{Cancel Reserve \%} = \frac{H}{H + S} \quad (12)$$

$$\text{Discount Amt \%} = \frac{D}{H + S} \quad (13)$$

$$content... \quad (14)$$

where  $H$  is the mean cancel reserve across the cohort,  $S$  is the mean seller advance amount across the cohort, and  $D$  is the mean discount amount across the cohort.

## Appendix A SQL Queries

### A.1 Bank Reconciliation Admin Details

```
SELECT ClearDate, Description, Notes, Amount as BnkAmt
FROM SPAdmin.dbo.SPA_SPBankingStat ORDER BY ClearDate ASC
```

```
SELECT ClearDate, Description, Notes, Account, Amount as PlugAmt
FROM SPAdmin.dbo.SPA_SPPlugStat ORDER BY ClearDate ASC
```

```
SELECT ClearDate, PaidTo as Payee, DisburseType as GroupName, PaymentAmount as DepAmt
FROM SPAdmin.dbo.SPA_SPDepositsStat ORDER BY ClearDate ASC
```

```
SELECT ClearDate, Vendor, GroupName, sum(PaymentAmount) as PaymntsAmt
FROM SPAdmin.dbo.SPA_SPPaymentsStat
GROUP BY Vendor, GroupName, ClearDate ORDER BY ClearDate, Vendor
```

```
SELECT ClearDate, sum(insurancereserve) as InsRsvAmt
FROM SPAdmin.dbo.SPA_Funded_Contracts GROUP BY cleardate;
```

### A.2 Bank Reconciliation Funder Summary and Details

```
SELECT * from SPAdmin.dbo.SPA_FundingBankStat ORDER BY ClearDate ASC;
```

```
SELECT TxDate, TxAmount, FTL.PosOrNegTx, TxDescription, FTL.PaidTo, FTL.PaidFrom
FROM dbo.SPF_Funding_Transaction_Log AS FTL
JOIN dbo.SPF_Funding_Transaction_Codes AS FTC
ON FTL.TxCODE = FTC.TxCODE
WHERE (FTL.CashTx = 1) AND (FTL.PolicyNumber IS NOT NULL);
```

### A.3 Scenario Modeling Summary and Details

The Scenario Modeling Info query:

```
SELECT distinct de.PolicyNumber, de.EffectiveDate, de.CancelDate, de.LastPaymentDate, de.IsCancel,
MAX(de.AmountFinanced) as AmountFinanced, MAX(sfd.TotalSalesPrice) as TotalSalePrice, MAX(de.LastPaymentDate) as LastPaymentDate,
MAX(CancelReserveAmount) as CancelReserveAmount, MAX(SellerAdvanceAmount) as SellerAdvanceAmount,
MAX(AdminPortionAmt) as AdminPortionAmt, MAX(InsReservePortionAmt) as InsReservePortionAmt,
MAX(de.CurrentInstallmentAmount) as CurrentInstallmentAmount, MAX(de.PaymentsMade) as PaymentsMade,
MAX(de.PaymentsMade+de.PaymentsRemaining) as Installments,
MAX(sfd.DownPayment) as DownPayment, MAX(de.ReturnedPremium) as ReturnedPremium, MAX(de.ReturnedPremium) as ReturnedPremium;
```

```

FROM dbo.daily_extract as de
JOIN  dbo.seller_funding_data as sfd on de.policynumber=sfd.policynumber
JOIN  dbo.admin_funding_data as afd on de.policynumber=afd.policynumber
JOIN  dbo.stoneeagle_all_customer_info as se on de.policynumber=se.policynumber
WHERE (de.PaymentsMade+de.PaymentsRemaining) = afd.installments
AND (de.PaymentsMade <= afd.installments)
GROUP BY de.PolicyNumber,de.EffectiveDate,de.CancelDate,de.LastPaymentDate,de.IsCancelled,de

```

The Scenario Modeling Info, more in-depth calculations used for calculating Holdback to be more efficient:

```

with scenario_info as (
SELECT distinct de.PolicyNumber,de.EffectiveDate,de.CancelDate,de.LastPaymentDate,de.IsCancelled,
MAX(de.AmountFinanced) as AmountFinanced,MAX(sfd.TotalSalesPrice) as TotalSalePrice,MAX(de.ReturnedPremium) as ReturnedPremium,
MAX(CancelReserveAmount) as CancelReserveAmount,MAX(SellerAdvanceAmount) as SellerAdvanceAmount,
MAX(AdminPortionAmt) as AdminPortionAmt,MAX(InsReservePortionAmt) as InsReservePortionAmt,
MAX(de.CurrentInstallmentAmount) as CurrentInstallmentAmount,MAX(de.PaymentsMade) as PaymentsMade,
MAX(de.PaymentsMade+de.PaymentsRemaining) as Installments,
MAX(sfd.DownPayment) as DownPayment,MAX(de.ReturnedPremium) as ReturnedPremium,MAX(de.ReturnedPremium) as ReturnedPremium,
FROM dbo.daily_extract as de
JOIN  dbo.seller_funding_data as sfd on de.policynumber=sfd.policynumber
JOIN  dbo.admin_funding_data as afd on de.policynumber=afd.policynumber
JOIN  dbo.stoneeagle_all_customer_info as se on de.policynumber=se.policynumber
WHERE (de.PaymentsMade+de.PaymentsRemaining) = afd.installments
AND (de.PaymentsMade <= afd.installments)
GROUP BY de.PolicyNumber,de.EffectiveDate,de.CancelDate,de.LastPaymentDate,de.IsCancelled,de
),

```

```

funding_fees as (
SELECT SunPathAccountingCode, Installments,DiscountPercentage,CancelPercentage,FlatCancelFee
FROM dbo.seller_info_funding_approved_partners as df1
JOIN  dbo.seller_info_funding_parameters as df3 on df1.SunPathSellerCode=df3.SunPathSellerCode
),

```

```

txcodes as (
SELECT distinct SunPathSellerCode,SunPathAccountingCode,SellerName
FROM dbo.daily_extract as t1
JOIN  dbo.seller_funding_data as t2
ON t1.PolicyNumber = t2.PolicyNumber
JOIN  dbo.seller_info_funding_approved_partners as t3
ON t1.sellercode = t3.paylinksellercode
),

```

```

rates as (
SELECT SellerName,

```

```

Installments,
CancelPercentage
FROM funding_fees
INNER JOIN txcodes on funding_fees.SunPathAccountingCode = txcodes.SunPathAccountingCode
),

info as (
SELECT PolicyNumber, CancelPercentage, EffectiveDate,TermDays,
CASE
WHEN (CancelDate is null) then LastPaymentDate
ELSE CancelDate
END as cancel_date
FROM scenario_info as t1
INNER JOIN rates as t2
ON t1.SellerName =t2.SellerName AND t1.PaymentsMade = t2.Installments
),

calculations as (
SELECT PolicyNumber,
CancelPercentage as rate,
datediff(day,EffectiveDate,cancel_date) as day_utilized,
datediff(day,EffectiveDate,cancel_date)/TermDays as VUR
FROM info
),

variables as (
SELECT t2.PolicyNumber,SellerName,IsCancelled,FundCo,Installments,
CurrentInstallmentAmount,PaymentsMade,ReturnedPremium,
DiscountAmount,CancelReserveAmount,SellerAdvanceAmount,
AmountFinanced,PaymentsRemaining,EffectiveDate,TermDays,SellerCost,
CASE
WHEN IsCancelled=1 OR PaymentsRemaining=0 then ReturnedPremium
WHEN IsCancelled=0 AND PaymentsRemaining!=0 then null
ELSE 0.0
END as end_contract_amt,
ROUND(rate*DiscountAmount,2) as prorated_fee,
day_utilized,VUR,(case when IsCancelled=1 then rate else 0.0 end) as rate,
CurrentInstallmentAmount * Installments as payment_plan_amount,
CurrentInstallmentAmount * PaymentsMade as total_install_rec,
ROUND((1-VUR)*AdminPortionAmt,2) as Amt_Owed_SPF_PreFee,
ROUND((1-VUR)*InsReservePortionAmt,2) as Amt_Owed_INS
FROM scenario_info as t1 inner JOIN calculations as t2
ON t1.PolicyNumber=t2.PolicyNumber
)

SELECT * FROM variables;

```

The Funding Fee Percents query:

```
SELECT SunPathAccountingCode, Installments,DiscountPercentage,
CancelPercentage,FlatCancelFee,ReservePercentage
FROM dbo.seller_info_funding_approved_partners as df1
JOIN dbo.seller_info_funding_parameters as df3
ON df1.SunPathSellerCode=df3.SunPathSellerCode;
```

The Daily Cashflows Transaction Log query:

```
SELECT
FTL.PolicyNumber,FTL.TxDate, FTC.TxDescription,
FTL.TxAmount * FTC.PosOrNegTx AS NetTransaction
FROM dbo.SPF_Funding_Transaction_Log AS FTL
LEFT JOIN dbo.SPF_Funding_Transaction_Codes AS FTC
ON FTL.TxCODE = FTC.TxCODE
WHERE (FTL.CashTx = 1) AND (FTL.PolicyNumber IS NOT NULL)
ORDER BY FTL.PolicyNumber, FTL.TxDate;
```