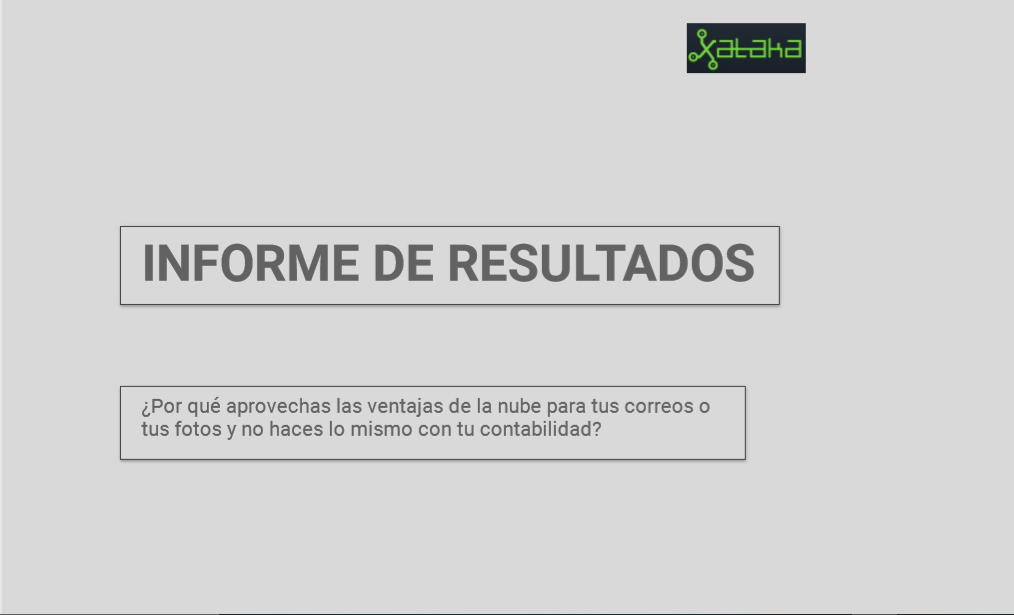
Name: **Hema Malini B**

Designation: **Data Analyst**

**Google Analytics Data Studio Reports**

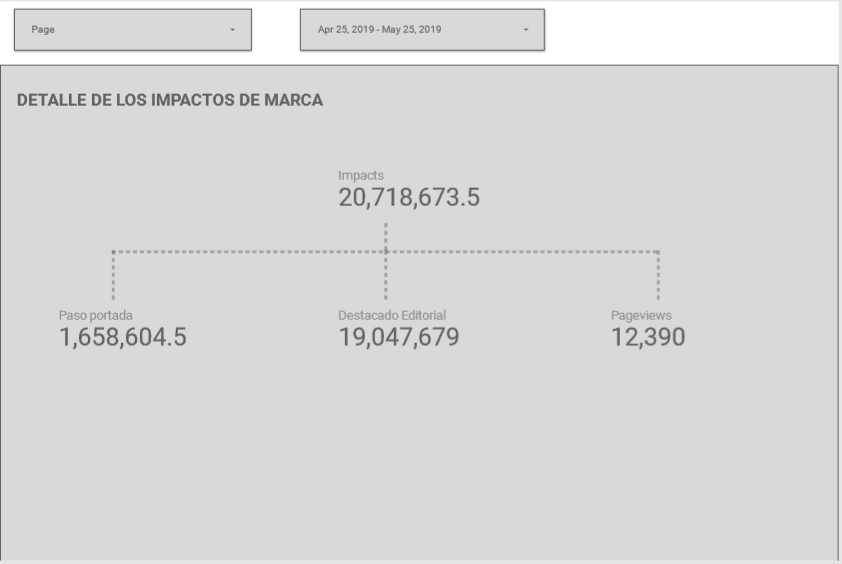
1. Xataka Data Studio Report
   1. Page 1



* 1. Page 2



* 1. Page 3



* 1. Page 4



* 1. Page 5



1. App Script Sample

*var startDate = '';*

*var pagePath = '';*

*var results = '';*

*var profileId = '';*

*var tableId = '';*

*var index = 0;*

*var sheet = '';*

*function runDemo() {*

*try {*

*clearSheets()*

*var data\_sheet = SpreadsheetApp.getActive().getSheetByName('Input');*

*for (index = 2; index<=10; ++index) {*

*check\_value = 'A'+index;*

*profileId = data\_sheet.getRange(check\_value).getValue();*

*if(profileId != '') {*

*results = getReportDataForProfilePP(index);*

*outputToSpreadsheetPP(index,results);*

*results = getReportDataForProfileDE(index);*

*outputToSpreadsheetDE(index,results);*

*/\*results = getReportDataForProfilePD(index);*

*outputToSpreadsheetPD(index,results);*

*results = getReportDataForProfileAVT(index);*

*outputToSpreadsheetPD(index,results);\*/*

*// Set Up the query arguments*

*check\_value = 'A'+index;*

*profileId = SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue();*

*var tableId = 'ga:' + profileId;*

*check\_value = 'B'+index;*

*var sDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*startDate = Utilities.formatDate(sDate, 'GMT+02:00', 'yyyy-MM-dd');*

*check\_value = 'C'+index;*

*var eDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*var endDate = Utilities.formatDate(eDate, 'GMT+02:00', 'yyyy-MM-dd');*

*check\_value = 'F'+index;*

*pagePath = SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue();*

*var metrics = ['ga:pageviews,ga:avgTimeOnPage,ga:bounceRate'];*

*var options = {*

*'dimensions': 'ga:PagePath',*

*'filters': 'ga:PagePath=@'+pagePath,*

*'sort': '-ga:pageviews',*

*//'segment': '',*

*'samplingLevel': 'HIGHER\_PRECISION',*

*'max-results': '5' // To limit the results to 5. Maximum number of results: 10000*

*}*

*// Fetch the report*

*var report = gaGet(tableId, startDate, endDate, metrics, options);*

*var data = report.rows;*

*var current\_sheet = SpreadsheetApp.getActive().getSheetByName('PageData');*

*// Get the range to write and write the results*

*var writeRange = current\_sheet.getRange(index, 1, data.length, data[0].length) // Read reference for getRange arguments*

*writeRange.setValues(data);*

*}*

*}*

*} catch(error) {*

*throw new Error(error);*

*}*

*}*

*function clearSheets() {*

*sheet = SpreadsheetApp.getActive().getSheetByName('PasoPorPortada');*

*sheet.clear();*

*sheet = SpreadsheetApp.getActive().getSheetByName('DestacadoEditorial');*

*sheet.clear();*

*sheet = SpreadsheetApp.getActive().getSheetByName('PageData');*

*sheet.clear();*

*sheet = SpreadsheetApp.getActive().getSheetByName('AverageTimeOnPage');*

*sheet.clear();*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Paso Por Portada Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*function getReportDataForProfilePP(index) {*

*var check\_value = '';*

*check\_value = 'A'+index;*

*profileId = SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue();*

*tableId = 'ga:' + profileId;*

*check\_value = 'B'+index;*

*var sDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*startDate = Utilities.formatDate(sDate, 'GMT+1:00', 'yyyy-MM-dd');*

*check\_value = 'D'+index;*

*var eDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*var endDate = Utilities.formatDate(eDate, 'GMT+1:00', 'yyyy-MM-dd');*

*var optArgs = {*

*'dimensions': '', // Comma separated list of dimensions.*

*'metrics': 'ga:pageviews',*

*'sort': 'ga:pageviews', // Sort by sessions descending, then keyword.*

*'filters': 'ga:pagepath==/;ga:country=@spain', // Display only google traffic.*

*'max-results': '250' // Display the first 250 results.*

*};*

*// Make a request to the API.*

*results = Analytics.Data.Ga.get(*

*tableId, // Table id (format ga:xxxxxx).*

*startDate, // Start-date (format yyyy-MM-dd).*

*endDate, // End-date (format yyyy-MM-dd).*

*'ga:pageviews', // Comma seperated list of metrics.*

*optArgs);*

*if (results.getRows()) {*

*return results;*

*} else {*

*throw new Error('No views (profiles) found');*

*}*

*}*

*function outputToSpreadsheetPP(index,results) {*

*var check\_value = '';*

*var resultVar = 0;*

*var sheet = SpreadsheetApp.getActive().getSheetByName('PasoPorPortada');*

*var cell = sheet.getRange("A1");*

*cell.setValue("Date");*

*var cell = sheet.getRange("B1");*

*cell.setValue("PasoPorPortado");*

*check\_value = 'A'+index;*

*var cell = sheet.getRange(check\_value);*

*cell.setValue(startDate);*

*resultVar = resultVar + results.getRows();*

*check\_value = 'B'+index;*

*var cell = sheet.getRange(check\_value);*

*cell.setValue(resultVar);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Paso Por Portada End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Destacado Editorial Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*function getReportDataForProfileDE(index) {*

*var check\_value = '';*

*check\_value = 'A'+index;*

*profileId = SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue();*

*var tableId = 'ga:' + profileId;*

*check\_value = 'B'+index;*

*var sDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*startDate = Utilities.formatDate(sDate, 'GMT+1:00', 'yyyy-MM-dd');*

*check\_value = 'E'+index;*

*var eDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*var endDate = Utilities.formatDate(eDate, 'GMT+1:00', 'yyyy-MM-dd');*

*var optArgs = {*

*'dimensions': '', // Comma separated list of dimensions.*

*'metrics': 'ga:pageviews',*

*'sort': 'ga:pageviews', // Sort by sessions descending, then keyword.*

*'filters': 'ga:country=@spain;ga:pagepath!@/quienes-somos;ga:pagepath!@/autor;ga:pagepath!@/respuestas',*

*'max-results': '250' // Display the first 250 results.*

*};*

*// Make a request to the API.*

*var results = Analytics.Data.Ga.get(*

*tableId, // Table id (format ga:xxxxxx).*

*startDate, // Start-date (format yyyy-MM-dd).*

*endDate, // End-date (format yyyy-MM-dd).*

*'ga:pageviews', // Comma seperated list of metrics.*

*optArgs);*

*if (results.getRows()) {*

*return results;*

*} else {*

*throw new Error('No views (profiles) found');*

*}*

*}*

*function outputToSpreadsheetDE(index,results) {*

*var resultVar = 0;*

*var check\_value = '';*

*var sheet = SpreadsheetApp.getActive().getSheetByName('DestacadoEditorial');*

*//sheet.clear();*

*var cell = sheet.getRange("A1");*

*cell.setValue("Date");*

*var cell = sheet.getRange("B1");*

*cell.setValue("DestacadoEditorial");*

*check\_value = 'A'+index;*

*var cell = sheet.getRange(check\_value);*

*cell.setValue(startDate);*

*resultVar = resultVar + results.getRows();*

*check\_value = 'B'+index;*

*var cell = sheet.getRange(check\_value);*

*cell.setValue(resultVar);*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Destacado Editorial End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Page Data Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*function getReportDataForProfilePD(index) {*

*var check\_value = '';*

*check\_value = 'A'+index;*

*profileId = SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue();*

*var tableId = 'ga:' + profileId;*

*check\_value = 'B'+index;*

*var sDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*startDate = Utilities.formatDate(sDate, 'GMT+02:00', 'yyyy-MM-dd');*

*check\_value = 'C'+index;*

*var eDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*var endDate = Utilities.formatDate(eDate, 'GMT+02:00', 'yyyy-MM-dd');*

*check\_value = 'F'+index;*

*pagePath = SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue();*

*var optArgs = {*

*'dimensions': 'ga:hostname', // Comma separated list of dimensions.*

*'metrics': 'ga:pageviews',*

*'sort': 'ga:pageviews', // Sort by sessions descending, then keyword.*

*'filters': 'ga:PagePath=@'+pagePath,*

*'max-results': '250' // Display the first 250 results.*

*};*

*// Make a request to the API.*

*var results = Analytics.Data.Ga.get(*

*tableId, // Table id (format ga:xxxxxx).*

*startDate, // Start-date (format yyyy-MM-dd).*

*endDate, // End-date (format yyyy-MM-dd).*

*'ga:pageviews', // Comma seperated list of metrics.*

*optArgs);*

*if (results.getRows()) {*

*return results;*

*} else {*

*throw new Error('No views (profiles) found');*

*}*

*}*

*function outputToSpreadsheetPD(index,results) {*

*var check\_value = '';*

*var sheet = SpreadsheetApp.getActive().getSheetByName('PageData');*

*var headerNames = [];*

*for (var i = 0, header; header = results.getColumnHeaders()[i]; ++i) {*

*headerNames.push(header.getName());*

*}*

*sheet.getRange(1, 1, 1, headerNames.length)*

*.setValues([headerNames]);*

*sheet.getRange(index, 1, results.getRows().length, headerNames.length)*

*.setValues( results.getRows());*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Page Data End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Average Time on Page Start \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*function getReportDataForProfileAVT(index) {*

*var check\_value = '';*

*check\_value = 'A'+index;*

*profileId = SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue();*

*var tableId = 'ga:' + profileId;*

*check\_value = 'B'+index;*

*var sDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*startDate = Utilities.formatDate(sDate, 'GMT+02:00', 'yyyy-MM-dd');*

*check\_value = 'C'+index;*

*var eDate = new Date(SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue());*

*var endDate = Utilities.formatDate(eDate, 'GMT+02:00', 'yyyy-MM-dd');*

*check\_value = 'F'+index;*

*pagePath = SpreadsheetApp.getActive().getSheetByName('Input').getRange(check\_value).getValue();*

*var optArgs = {*

*'dimensions': 'ga:PagePath', // Comma separated list of dimensions.*

*'metrics': 'ga:avgTimeOnPage',*

*'sort': 'ga:avgTimeOnPage', // Sort by sessions descending, then keyword.*

*'filters': 'ga:PagePath=@'+pagePath,*

*'max-results': '250' // Display the first 250 results.*

*};*

*// Make a request to the API.*

*var results = Analytics.Data.Ga.get(*

*tableId, // Table id (format ga:xxxxxx).*

*startDate, // Start-date (format yyyy-MM-dd).*

*endDate, // End-date (format yyyy-MM-dd).*

*'ga:avgTimeOnPage', // Comma seperated list of metrics.*

*optArgs);*

*if (results.getRows()) {*

*return results;*

*} else {*

*throw new Error('No views (profiles) found');*

*}*

*}*

*function outputToSpreadsheetAVT(index,results) {*

*var check\_value = '';*

*var sheet = SpreadsheetApp.getActive().getSheetByName('AverageTimeOnPage');*

*var headerNames = [];*

*for (var i = 0, header; header = results.getColumnHeaders()[i]; ++i) {*

*headerNames.push(header.getName());*

*}*

*sheet.getRange(1, 1, 1, headerNames.length)*

*.setValues([headerNames]);*

*sheet.getRange(index, 1, results.getRows().length, headerNames.length)*

*.setValues( results.getRows());*

*}*

*/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Page Data End \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/*

*function gaGet(tableId, startDate, endDate, metrics, options) {*

*// Apply standard options*

*options = options || {};*

*options['max-results'] = options['max-results'] || '10000';*

*// If errors persist up to 5 times then terminate the program.*

*for (var i = 0; i < 5; i++) {*

*try {*

*return Analytics.Data.Ga.get(tableId, startDate, endDate, metrics, options); // 503*

*} catch (err) {*

*// https://developers.google.com/analytics/devguides/reporting/core/v3/coreErrors*

*if (err.message.indexOf('a server error occurred') > -1) {*

*Logger.log('Backend Error');*

*// Note: Don't listen to Google's reply and retry request after 2 minutes*

*Utilities.sleep(2 \* 60 \* 1000);*

*} else if (err.message.indexOf('User Rate') > -1) {*

*Logger.log('Rate Limit Error');*

*// Exponential Backoff*

*Utilities.sleep(1000 \* Math.pow((i + 1), 2));*

*} else if (err.message.indexOf('too many concurrent connections') > -1) {*

*Logger.log('Concurrent Connections Error');*

*// Exponential Backoff*

*Utilities.sleep(1000 \* Math.pow((i + 1), 2));*

*} else {*

*Logger.log(err);*

*throw err;*

*}*

*}*

*}*

*throw 'Error. Max retries reached';*

*}*

**PYTHON SPSS SAMPLES**

1. Code Sample

*\* Encoding: UTF-8.*

*\* Encoding: .*

*\* Encoding: .*

*\* Encoding: .*

*BEGIN PROGRAM PYTHON3.*

*import spss, datetime as dt*

*v3 = []*

*v4 = []*

*v5 = []*

*v6 = []*

*modifier = []*

*i = 1*

*spss.Submit("INSERT FILE='C:/PPA/MASTER\_IMPORT\_GENERIC\_UNICODE\_PROTO.sps'.")*

*spss.Submit("!MASTER\_INPUT group = 11 rvu\_year = '17' data\_end = '123117' price\_end = '123117' generic\_limit = 99 .")*

*from savReaderWriter import \**

*with SavReader('C:/PPA/G11\_SETUP\_LABELS\_ALL\_ACTIVE.SAV', returnHeader=True) as reader:*

*header = reader.\_\_next\_\_()*

*for line in reader:*

*v3\_append = str((line[1]).decode('utf-8')) + '\_line\_' + str(i)*

*v3.append(v3\_append)*

*v4\_append = str((line[2]).decode('utf-8')) + '\_line\_' + str(i)*

*v4.append(v4\_append)*

*v5\_append = str((line[9]).decode('utf-8')) + '\_line\_' + str(i)*

*v5.append(v5\_append)*

*v6\_append = str((line[6]).decode('utf-8')) + '\_line\_' + str(i)*

*v6.append(v6\_append)*

*modifier\_append = str((line[14]).decode('utf-8')) + '\_line\_' + str(i)*

*modifier.append(modifier\_append)*

*i = i + 1*

*from datetime import datetime*

*try:*

*import tkinter as tk*

*from tkinter import ttk*

*except ImportError:*

*import Tkinter as tk*

*import ttk*

*from tkcalendar import Calendar, DateEntry*

*def run():*

*global v3\_in,v4\_in,v5\_in,v6\_in,modifier\_in, v3\_entry\_in, v4\_entry\_in, v5\_entry\_in, v6\_entry\_in*

*v3\_in = (str\_v3.get()).split('\_line\_')[1]*

*v4\_in = (str\_v4.get()).split('\_line\_')[1]*

*v5\_in = (str\_v5.get()).split('\_line\_')[1]*

*v6\_in = (str\_v6.get()).split('\_line\_')[1]*

*modifier\_in = (str\_modifier.get()).split('\_line\_')[1]*

*v3\_entry\_in = v3\_entry.get()*

*v4\_entry\_in = v4\_entry.get()*

*v5\_entry\_in = v5\_entry.get()*

*v6\_entry\_in = v6\_entry.get()*

*modifier\_entry\_in = modifier\_entry.get()*

*spss.Submit("GET FILE='C:/PPA/G11\_SETUP\_LABELS\_ALL\_ACTIVE.SAV'.")*

*spss.StartDataStep()*

*datasetObj = spss.Dataset()*

*v3\_in = int(v3\_in) - 1*

*datasetObj.cases[v3\_in,3] = str(v3\_entry\_in)*

*v4\_in = int(v4\_in) - 1*

*datasetObj.cases[v4\_in,4] = str(v4\_entry\_in)*

*v5\_in = int(v5\_in) - 1*

*datasetObj.cases[v5\_in,10] = str(v5\_entry\_in)*

*v6\_in = int(v6\_in) - 1*

*datasetObj.cases[v6\_in,7] = str(v6\_entry\_in)*

*modifier\_in = int(modifier\_in) - 1*

*datasetObj.cases[modifier\_in,15] = float(modifier\_entry\_in)*

*str\_len = len(datasetObj.cases)*

*for j in range(0, str\_len):*

*datasetObj.cases[0,3] = '123'*

*datasetObj.cases[0,4] = '123'*

*datasetObj.cases[0,10] = '123'*

*datasetObj.cases[0,7] = '123'*

*datasetObj.cases[0,15] = 2*

*mainframe.destroy()*

*spss.EndDataStep()*

*spss.Submit("SAVE OUTFILE='C:/PPA/G11\_SETUP\_LABELS\_ALL\_ACTIVE.SAV'.")*

*spss.Submit("INSERT FILE='C:/PPA/MASTER\_UPDATE\_GENERIC\_UNICODE\_PROTO.sps'.")*

*spss.Submit("!MASTER\_UPDATE group = 11 rvu\_year = '17' data\_end = '123117' price\_end = '123117' . ")*

*def cancel():*

*mainframe.destroy()*

*exit()*

*root = tk.Toplevel()*

*root.withdraw()*

*root.geometry('1000x500')*

*root.resizable(width=False, height=False)*

*root.rowconfigure(0, weight=0)*

*root.columnconfigure(0, weight=1)*

*root.update()*

*# Create a Tkinter variable*

*mainframe = tk.Frame(root)*

*mainframe.grid(column=0,row=0, sticky=('N','W','E','S') )*

*mainframe.columnconfigure(0, weight = 1)*

*mainframe.rowconfigure(0, weight = 1)*

*mainframe.pack(pady = 100, padx = 100)*

*str\_v3 = tk.StringVar(root)*

*popupMenu\_v3 = tk.OptionMenu(mainframe, str\_v3, \*v3)*

*tk.Label(mainframe, text="Select Primary Provider").grid(row = 1, column = 1)*

*popupMenu\_v3.grid(row = 1, column =2)*

*# on change dropdown value*

*def change\_dropdown(\*args):*

*print( str\_v3.get() )*

*# link function to change dropdown*

*str\_v3.trace('w', change\_dropdown)*

*tk.Label(mainframe, text='CURRENT\_PROVIDER\_LABEL', fg="black", font=("Courier", 10)).grid(row=1, column=5)*

*v3\_entry = tk.Entry(mainframe, text='', width=20)*

*v3\_entry.grid(row=1, column=6)*

*str\_v4 = tk.StringVar(root)*

*popupMenu\_v4 = tk.OptionMenu(mainframe, str\_v4, \*v4)*

*tk.Label(mainframe, text="Select Supervising Provider").grid(row = 2, column = 1)*

*popupMenu\_v4.grid(row = 2, column =2)*

*# on change dropdown value*

*def change\_dropdown(\*args):*

*print( str\_v4.get() )*

*# link function to change dropdown*

*str\_v4.trace('w', change\_dropdown)*

*tk.Label(mainframe, text='CURRENT\_SUPERVISOR\_LABEL', fg="black", font=("Courier", 10)).grid(row=2, column=5)*

*v4\_entry = tk.Entry(mainframe, text='', width=20)*

*v4\_entry.grid(row=2, column=6)*

*str\_v5 = tk.StringVar(root)*

*popupMenu\_v5 = tk.OptionMenu(mainframe, str\_v5, \*v5)*

*tk.Label(mainframe, text="Select Location of Service").grid(row = 3, column = 1)*

*popupMenu\_v5.grid(row = 3, column =2)*

*# on change dropdown value*

*def change\_dropdown(\*args):*

*print( str\_v5.get() )*

*# link function to change dropdown*

*str\_v5.trace('w', change\_dropdown)*

*tk.Label(mainframe, text='CURRENT\_LOCATION\_LABEL', fg="black", font=("Courier", 10)).grid(row=3, column=5)*

*v5\_entry = tk.Entry(mainframe, text='', width=20)*

*v5\_entry.grid(row=3, column=6)*

*str\_v6 = tk.StringVar(root)*

*popupMenu\_v6 = tk.OptionMenu(mainframe, str\_v6, \*v6)*

*tk.Label(mainframe, text="Select Primary Insurance").grid(row = 4, column = 1)*

*popupMenu\_v6.grid(row = 4, column =2)*

*# on change dropdown value*

*def change\_dropdown(\*args):*

*print( str\_v6.get() )*

*# link function to change dropdown*

*str\_v6.trace('w', change\_dropdown)*

*tk.Label(mainframe, text='CURRENT\_INSURANCE\_LABEL', fg="black", font=("Courier", 10)).grid(row=4, column=5)*

*v6\_entry = tk.Entry(mainframe, text='', width=20)*

*v6\_entry.grid(row=4, column=6)*

*str\_modifier = tk.StringVar(root)*

*popupMenu\_modifier = tk.OptionMenu(mainframe, str\_modifier, \*modifier)*

*tk.Label(mainframe, text="Select Modifier").grid(row = 5, column = 1)*

*popupMenu\_modifier.grid(row = 5, column =2)*

*# on change dropdown value*

*def change\_dropdown(\*args):*

*print( str\_modifier.get() )*

*# link function to change dropdown*

*str\_modifier.trace('w', change\_dropdown)*

*tk.Label(mainframe, text='CURRENT\_MODIFIER\_WEIGHT', fg="black", font=("Courier", 10)).grid(row=5, column=5)*

*modifier\_entry = tk.Entry(mainframe, text='', width=20)*

*modifier\_entry.grid(row=5, column=6)*

*run\_button = ttk.Button(mainframe, text="SUBMIT", command=run)*

*run\_button.grid(row=6, column=3, sticky=('N','W','E','S'))*

*cancel\_button = ttk.Button(mainframe, text="CANCEL", command=cancel)*

*cancel\_button.grid(row=6, column=4, sticky=('N','W','E','S'))*

*mainframe.grid\_columnconfigure(6, minsize=120)*

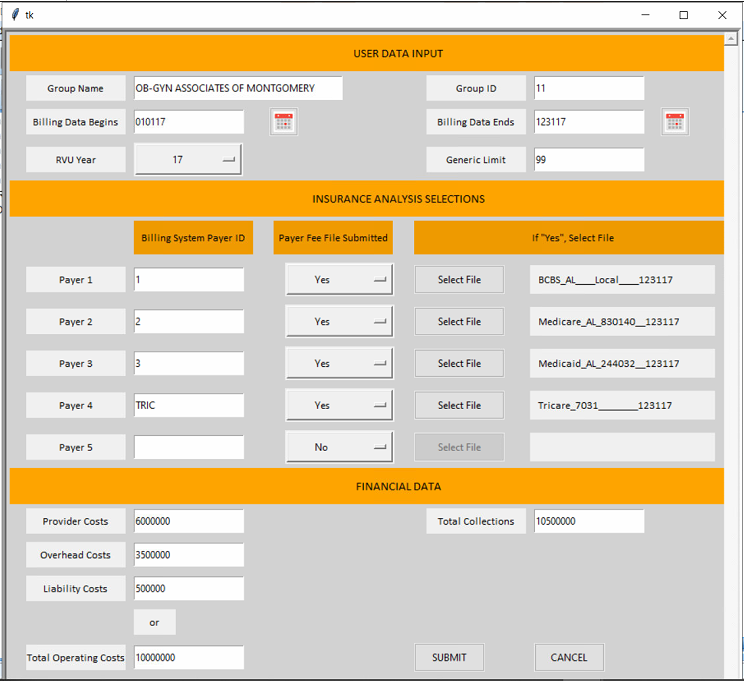
*root.wm\_deiconify()*

*root.mainloop()*

*print()*

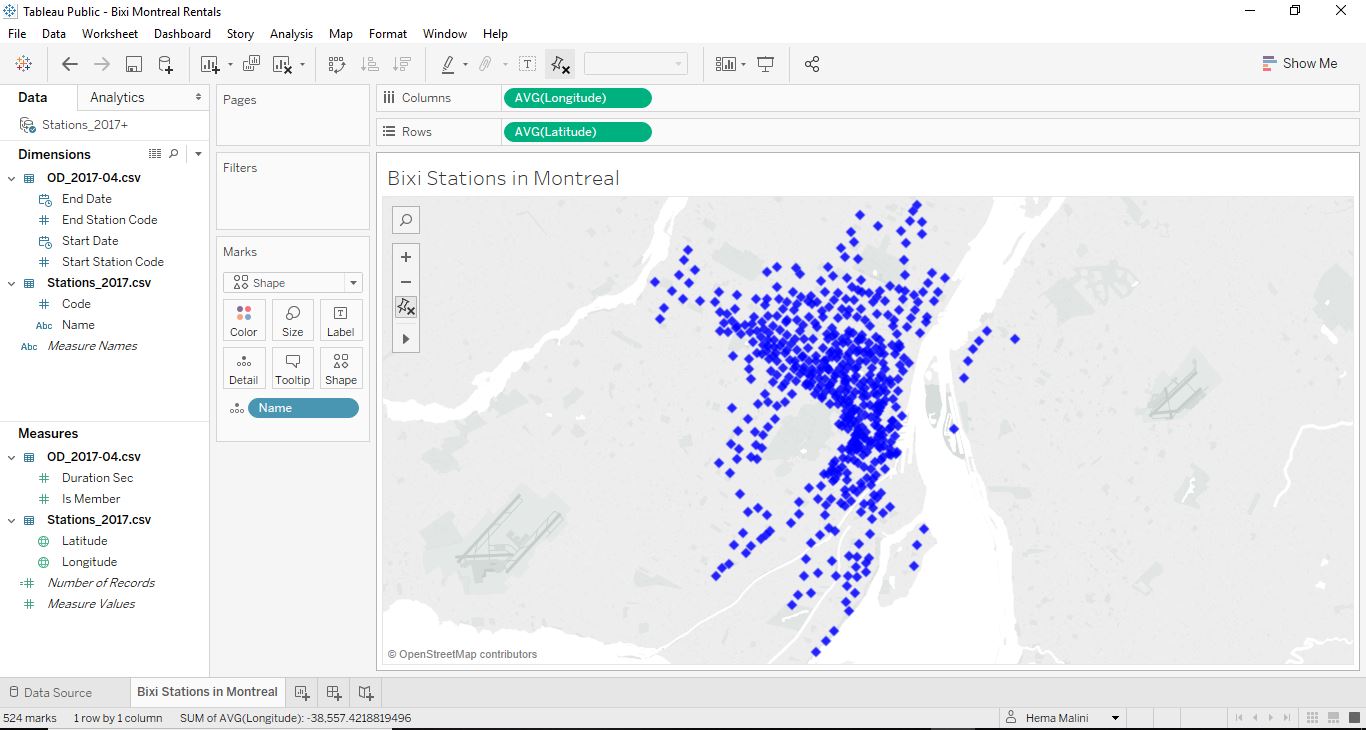
*END PROGRAM.*

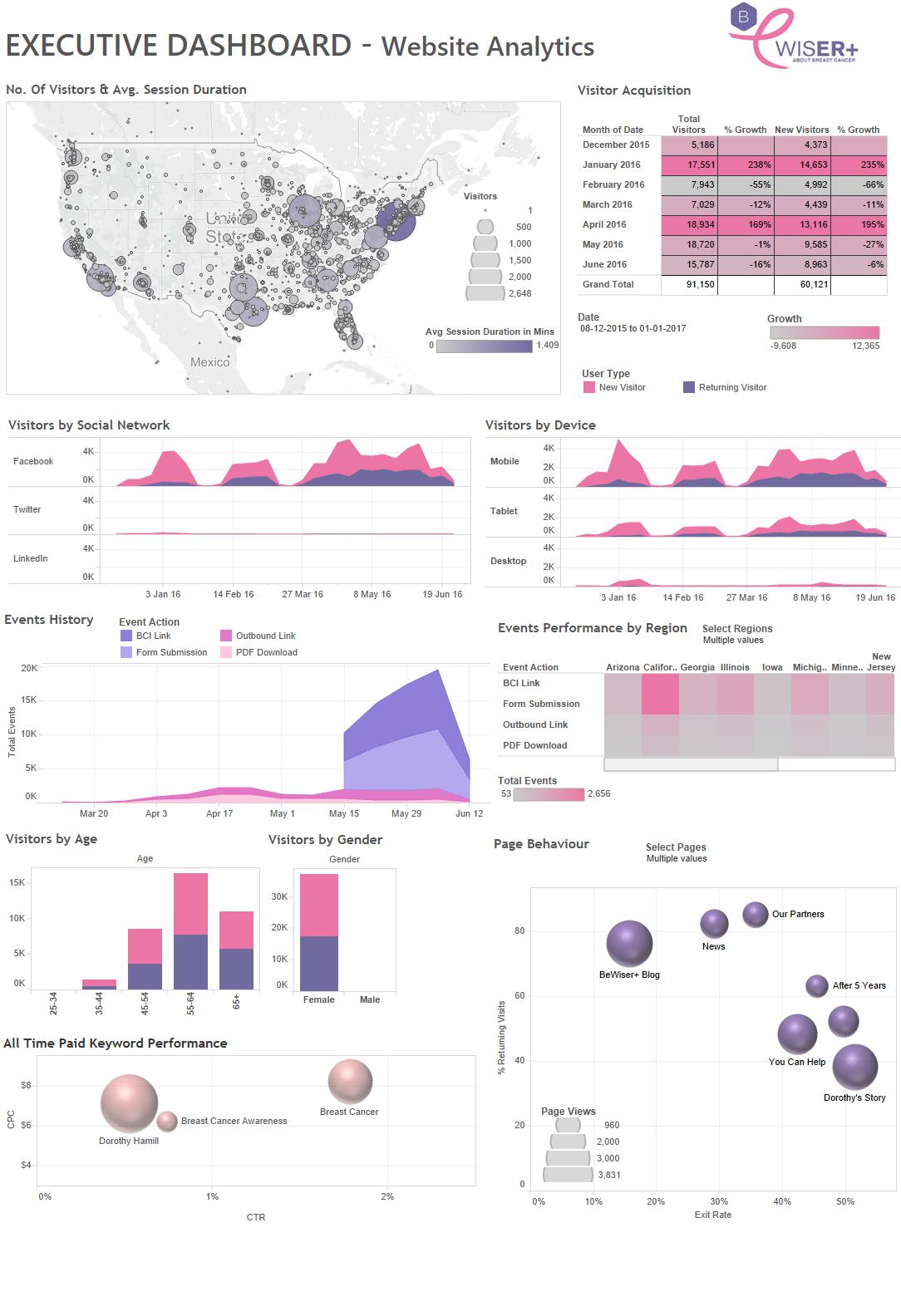
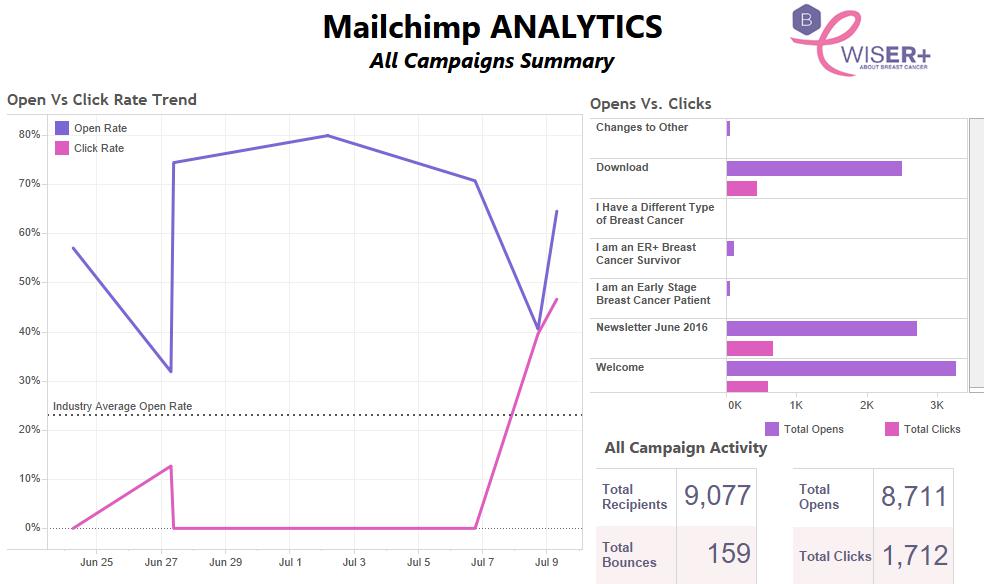
1. Output Sample

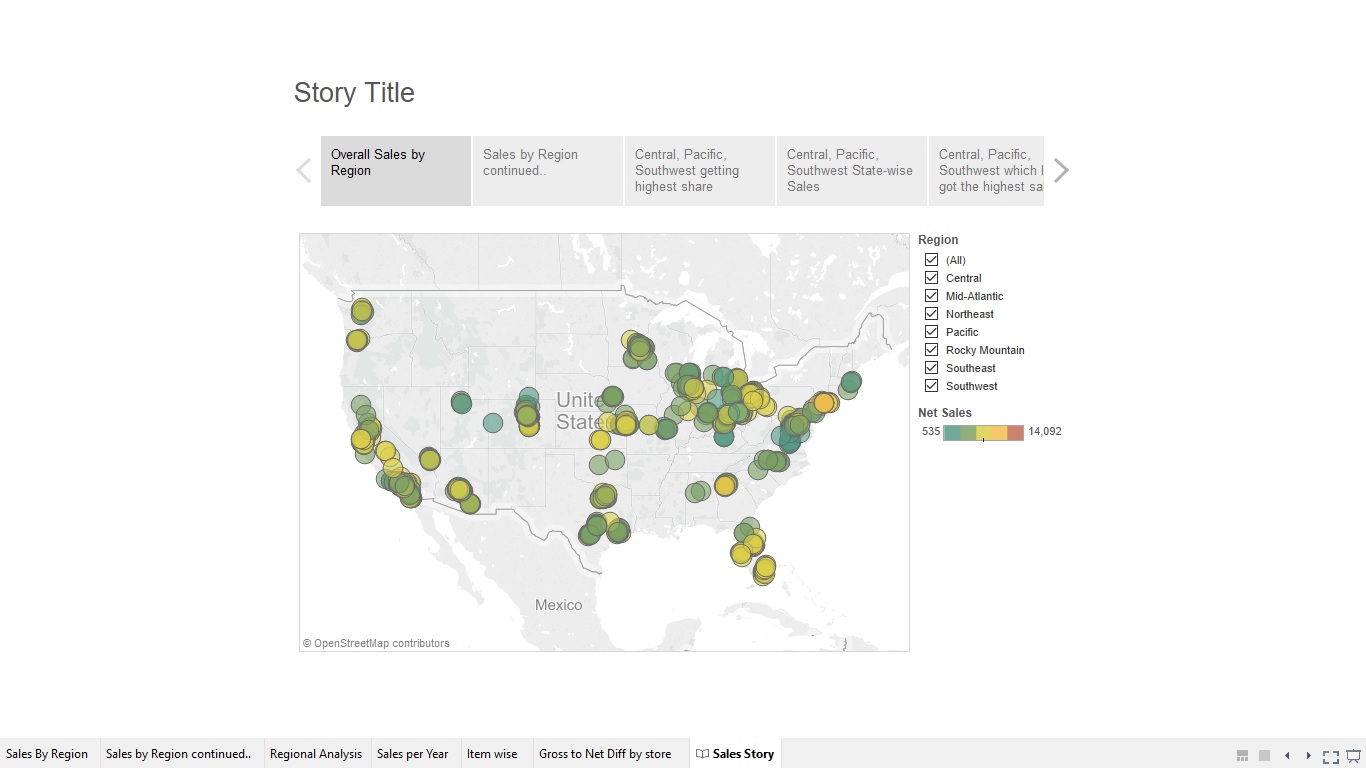


**TABLEAU WORK SAMPLES**

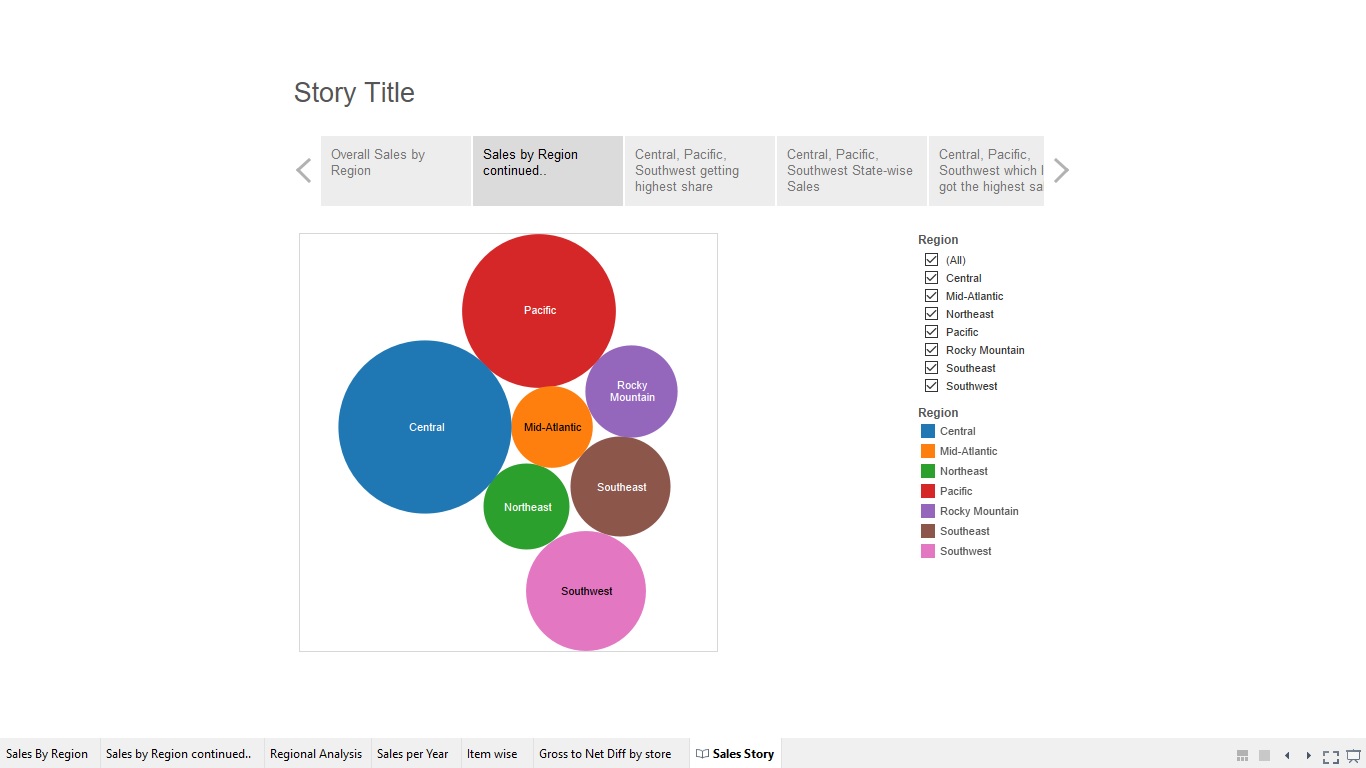
1. Bixi Montreal Rentals: Daily bike rentals Viz



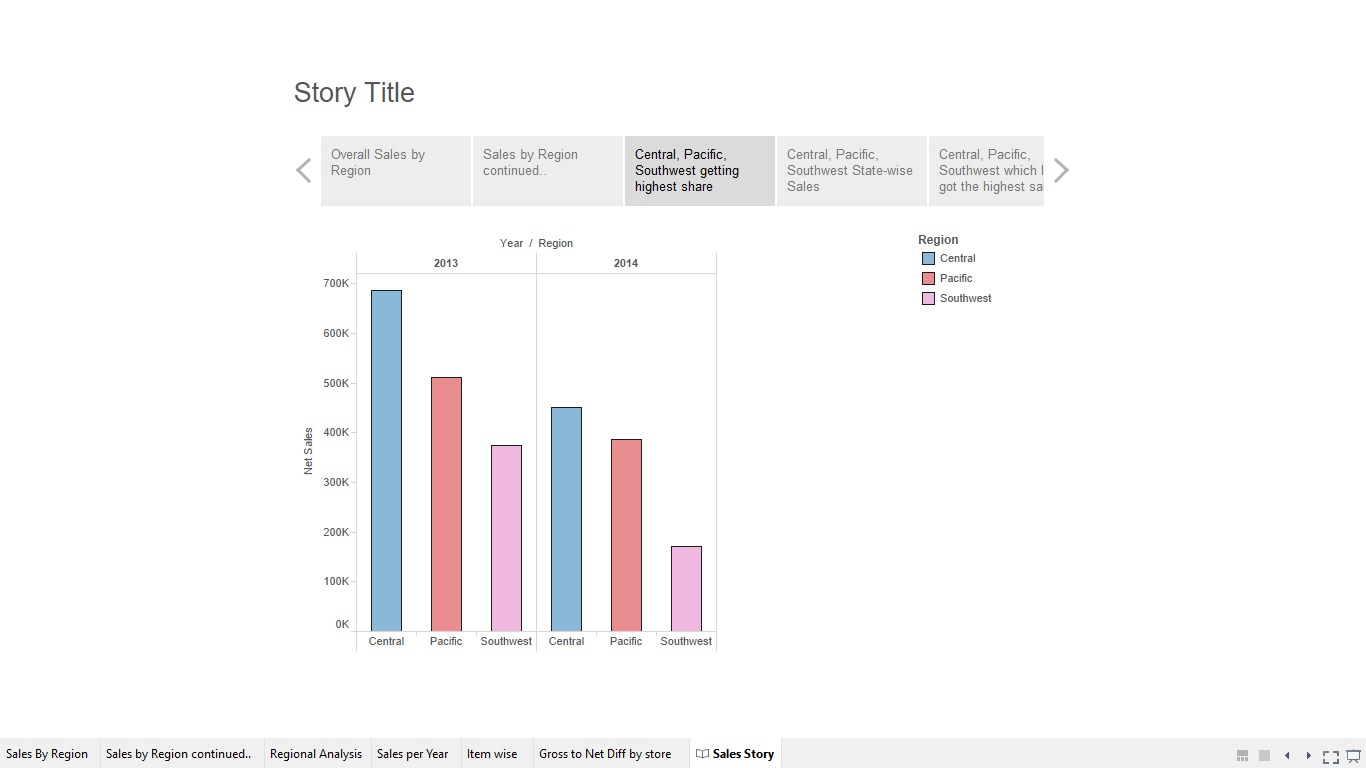
1. Be Wiser Cancer Project; Excel Connection, Website Analytics, Visitors by Device, Visitors by social network, Visitor Acquisition and other event, page and keyword analytics.
2. Be Wiser Cancer Project; Excel Connection, MailChimp Analytics, Campaigns Summary & Activity, Open & Click Rate Statistics
3. Guru Sales Project Story; Excel Connection, Sales Across all Regions



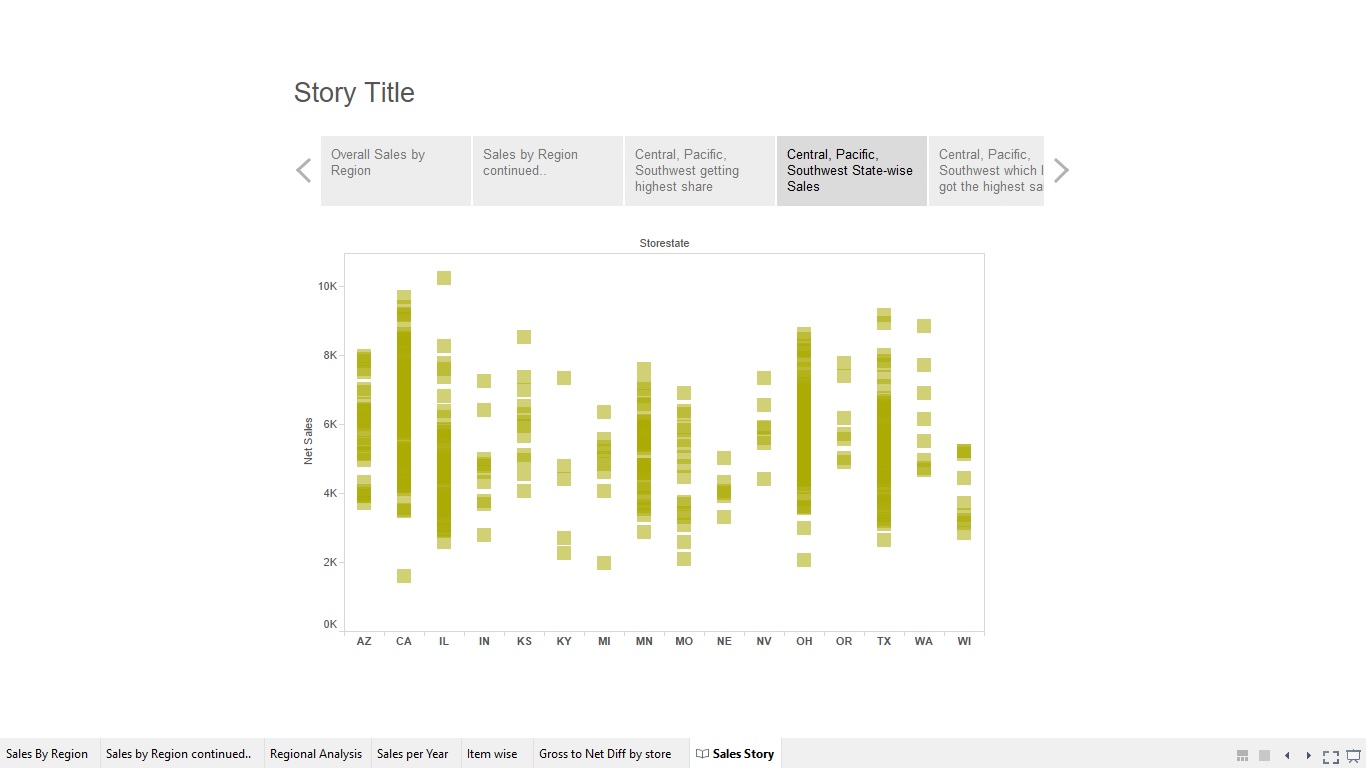
* 1. Sales by region in bubble



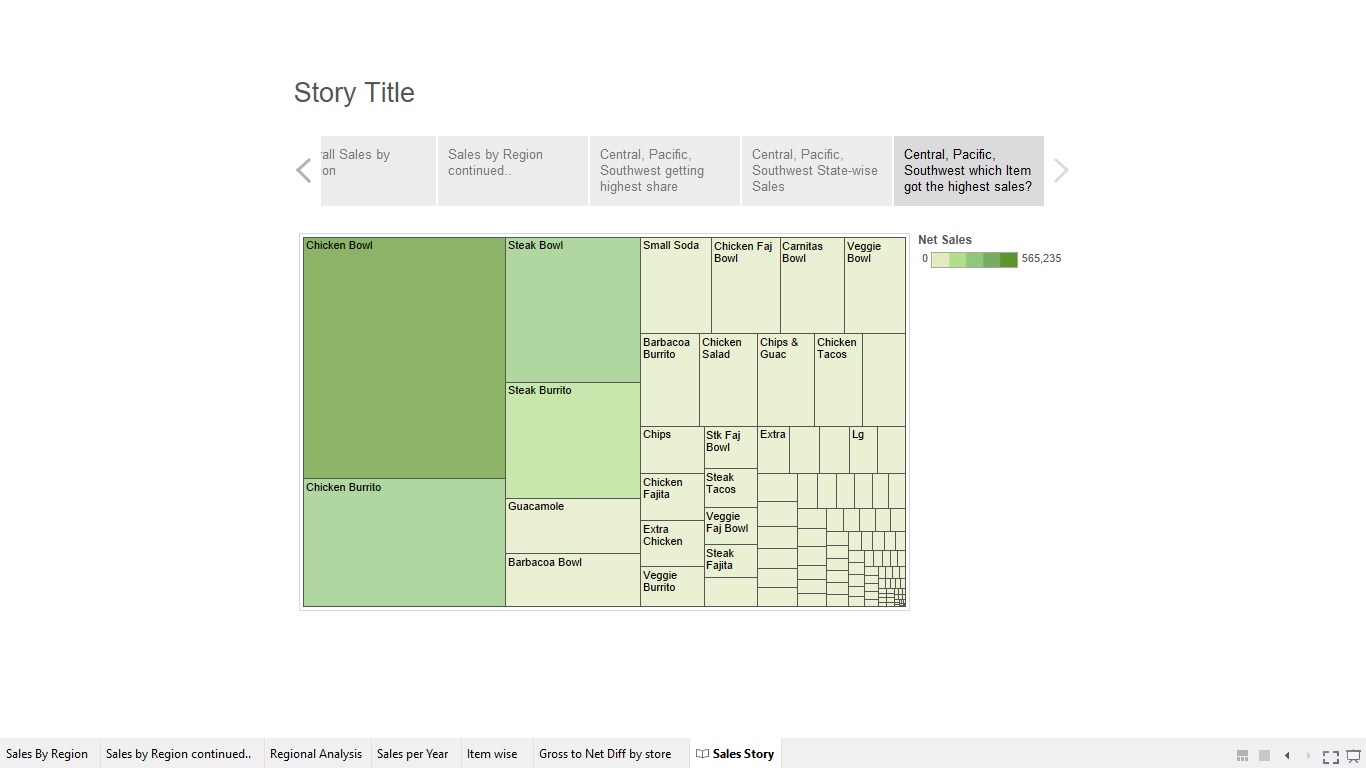
* 1. Sales share by region and year



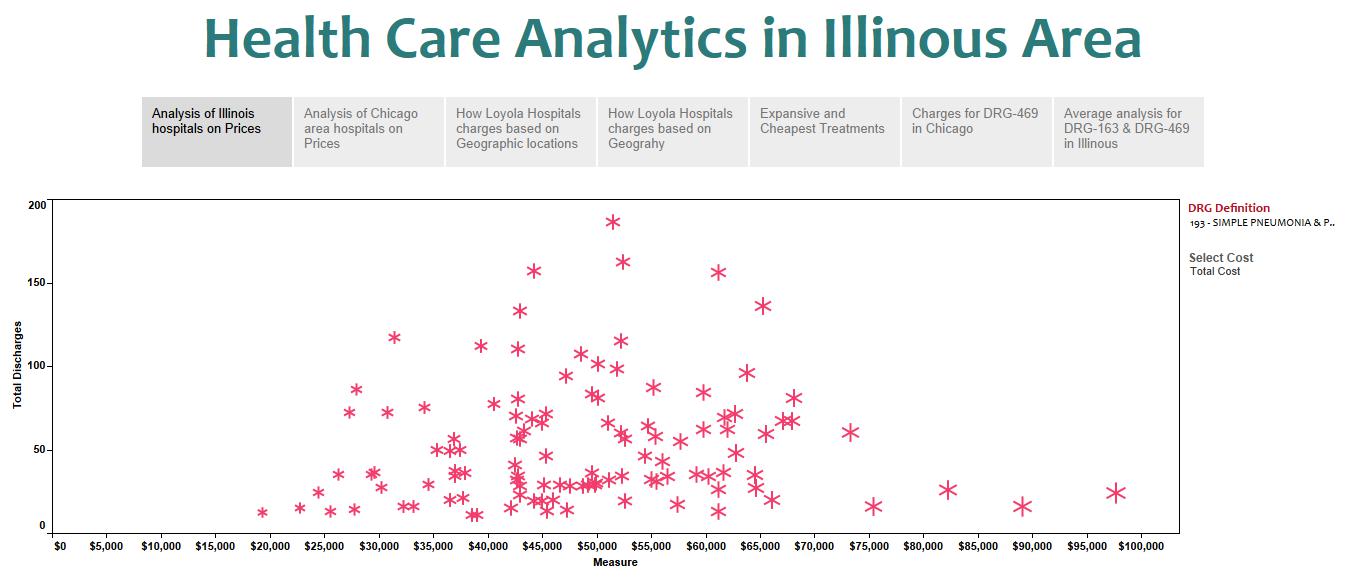
* 1. State-wise Sales



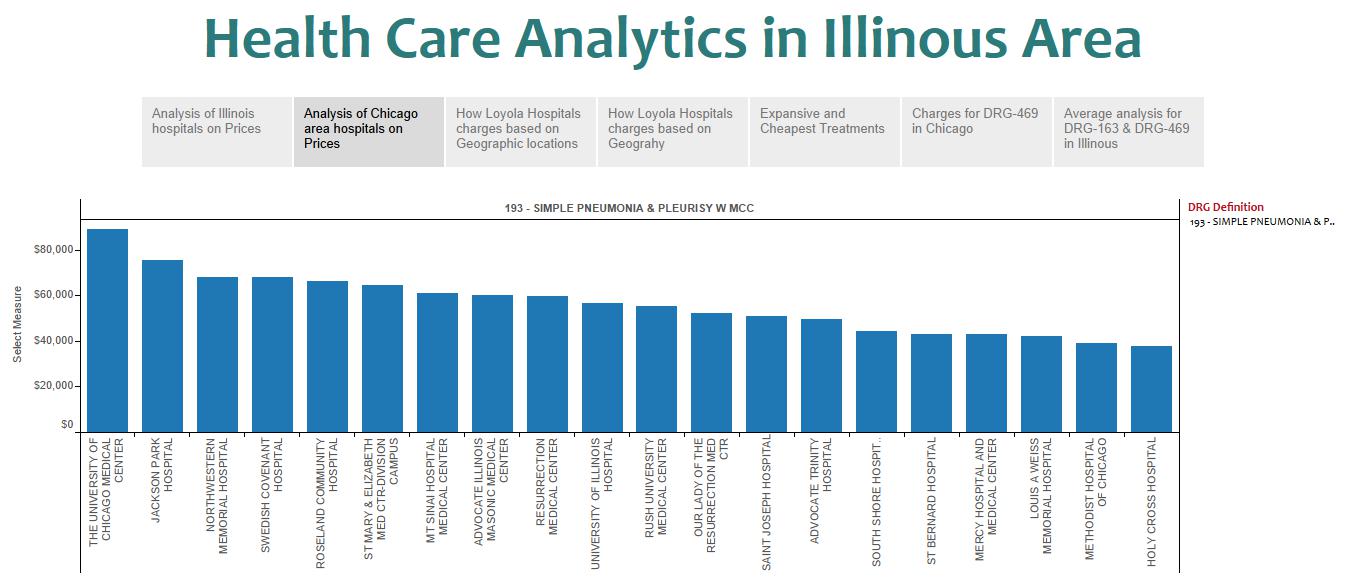
* 1. Product-wise sales



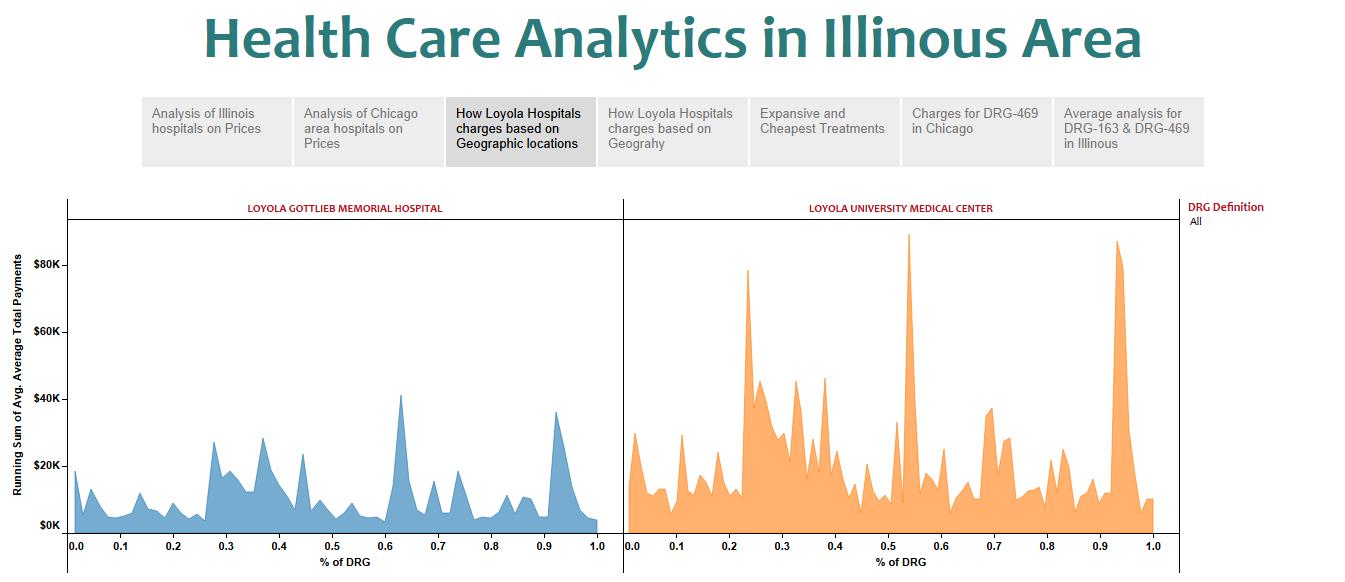
1. Health Care Project Story; Excel Connection, Price Analysis by Hospital



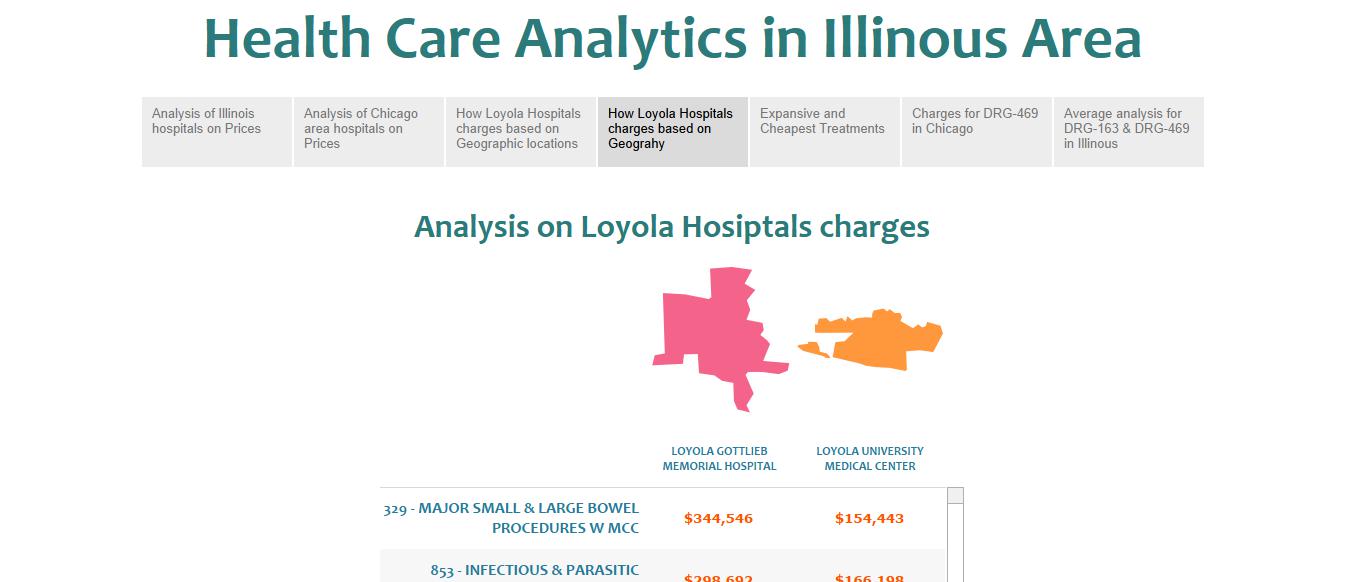
* 1. Price Analysis by Area



* 1. Hospital Price Comparison



* 1. Charges based on geography



* 1. Treatment Cost



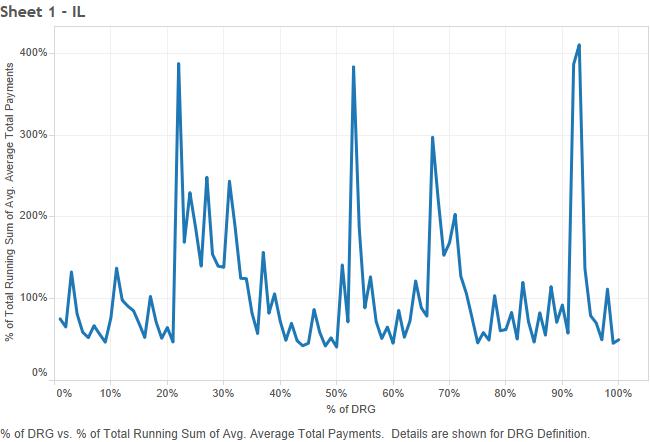
* 1. Drug-wise Charges

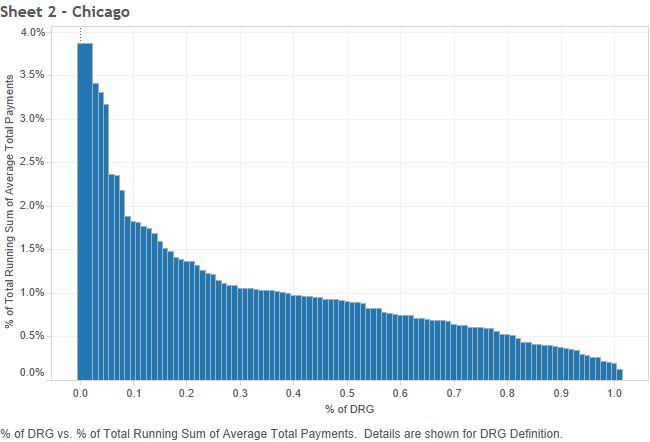


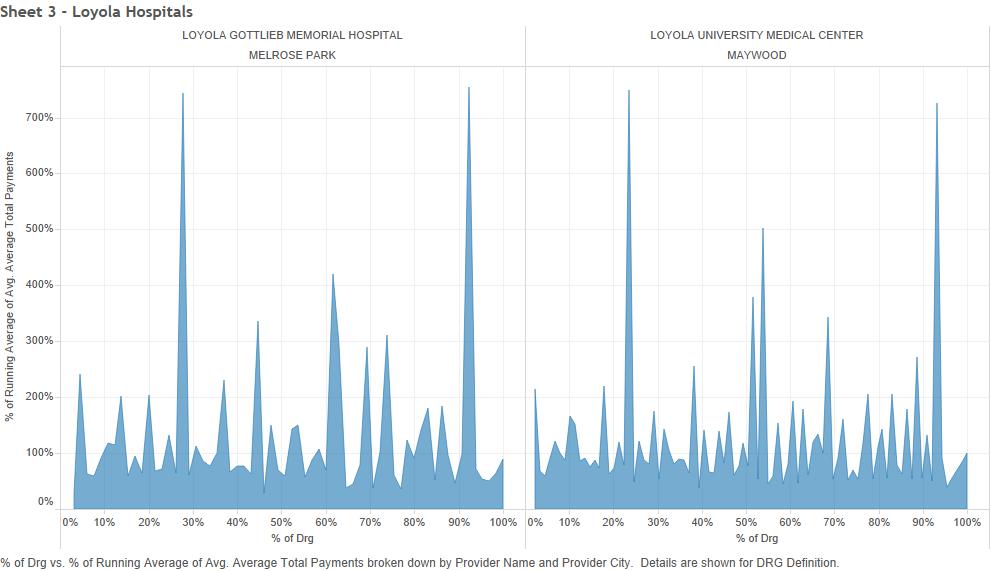
* 1. Drug Analysis



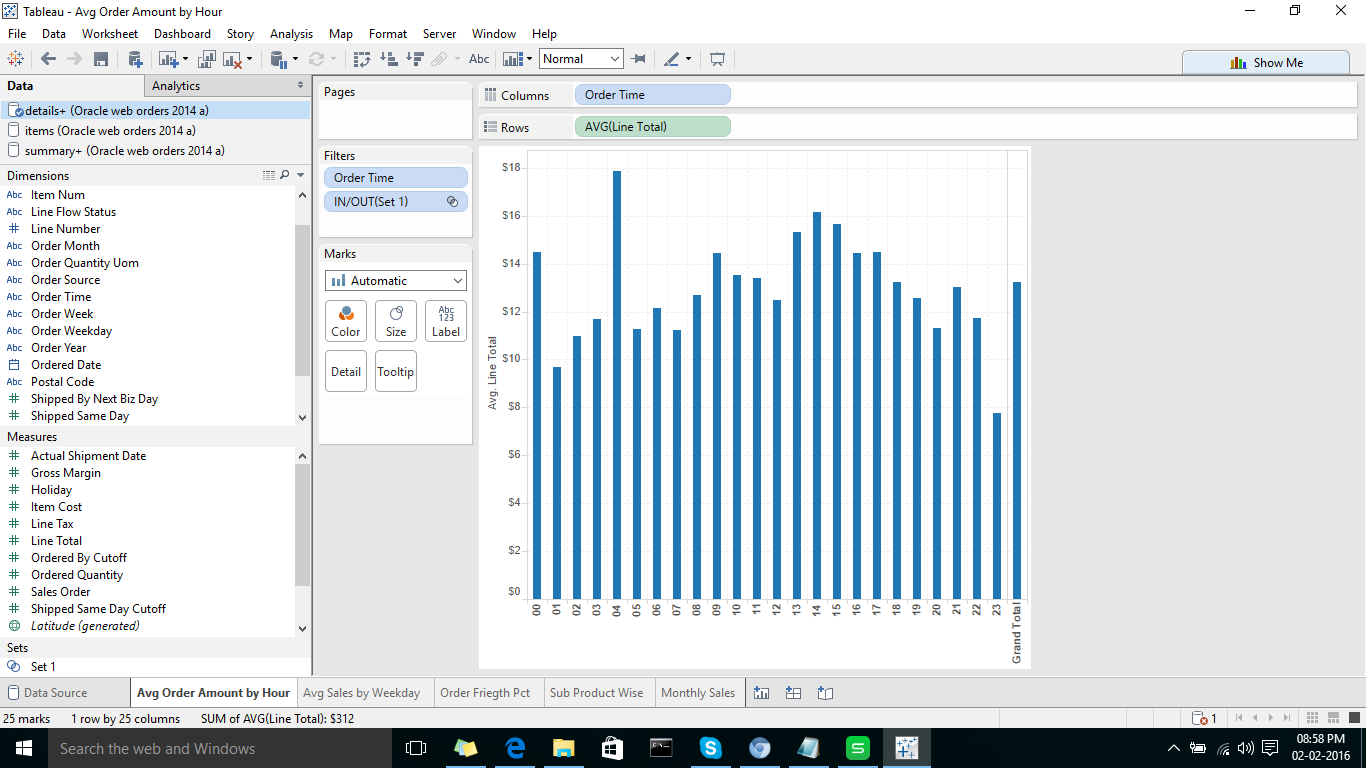
* 1. Health Care Individuals



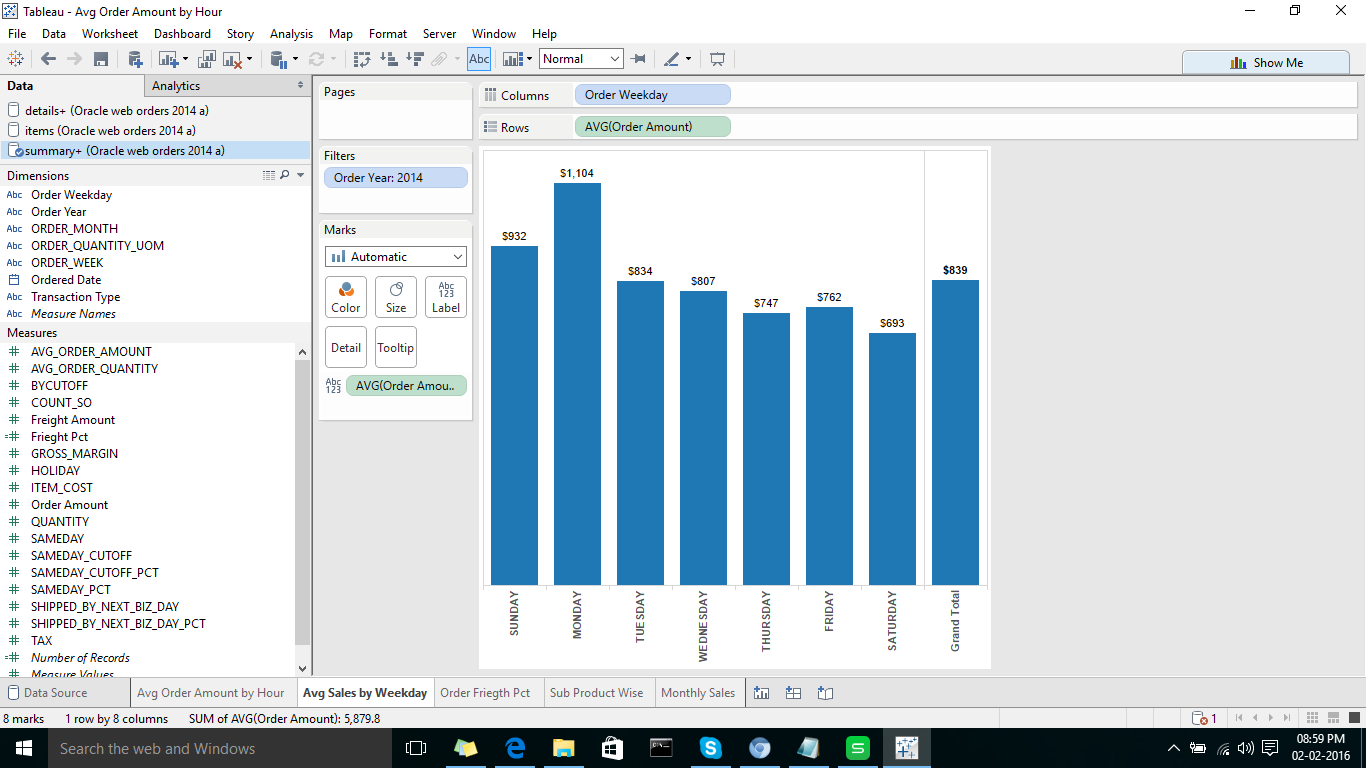




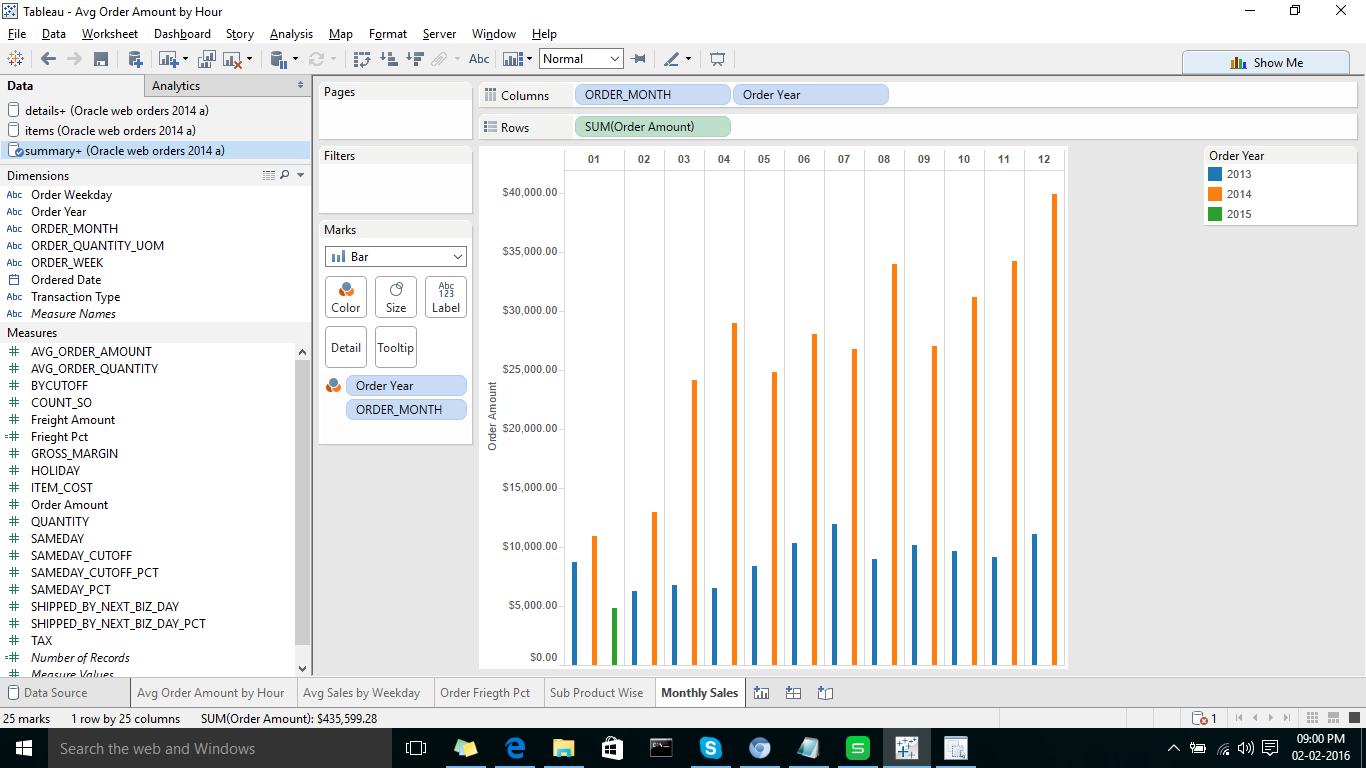
1. Oracle Tableau Project; Oracle Data Source, Order Amount per HourSales by weekday



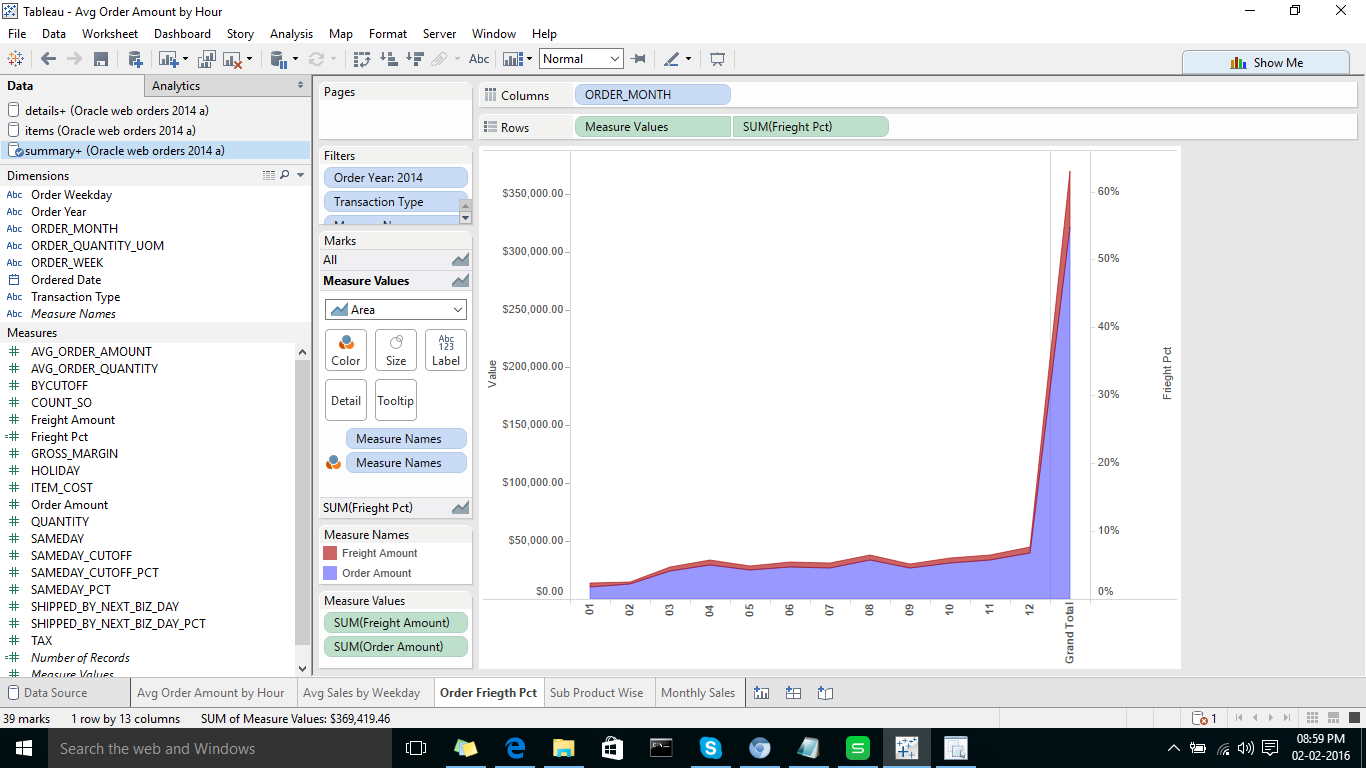
* 1. Sales by Weekday



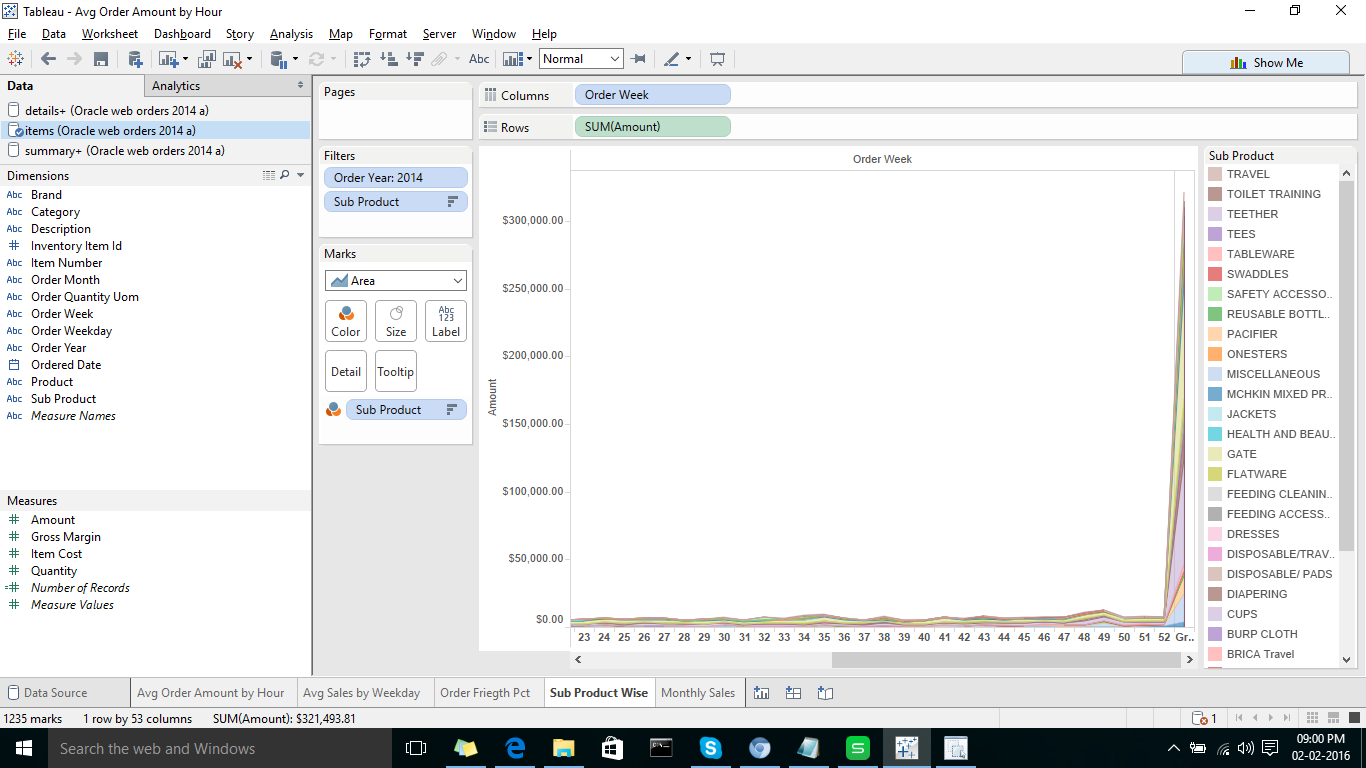
* 1. Monthly Sales



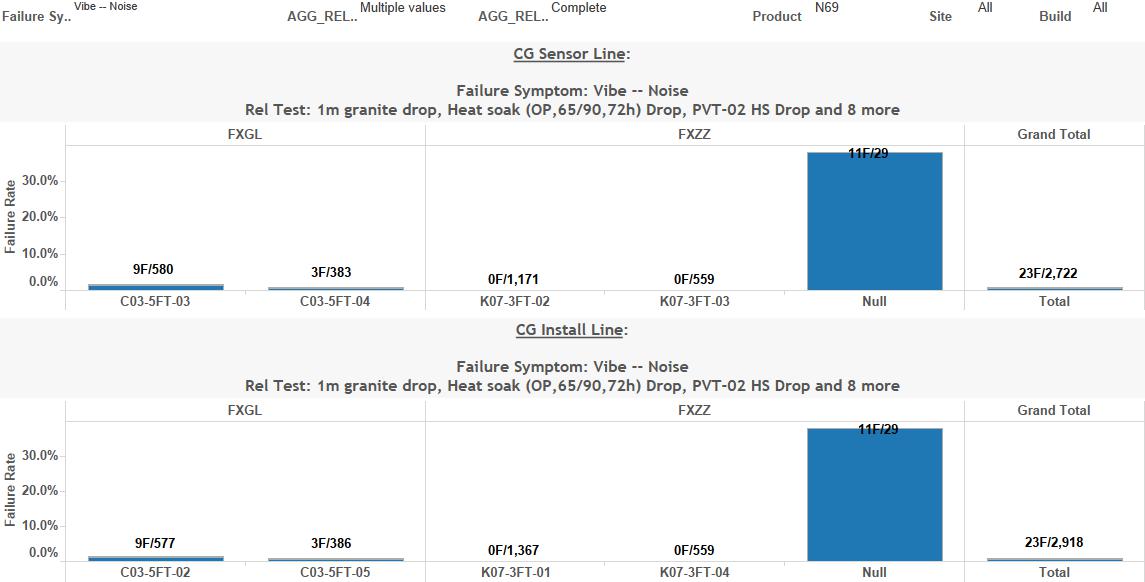
* 1. Order Freight Percent



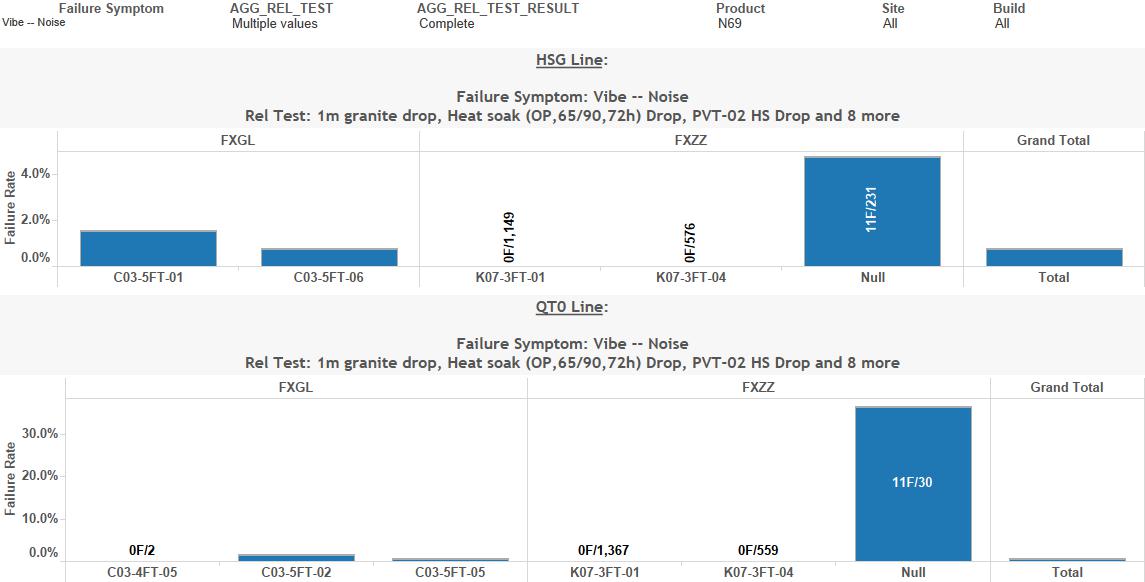
* 1. Sub-product Wise



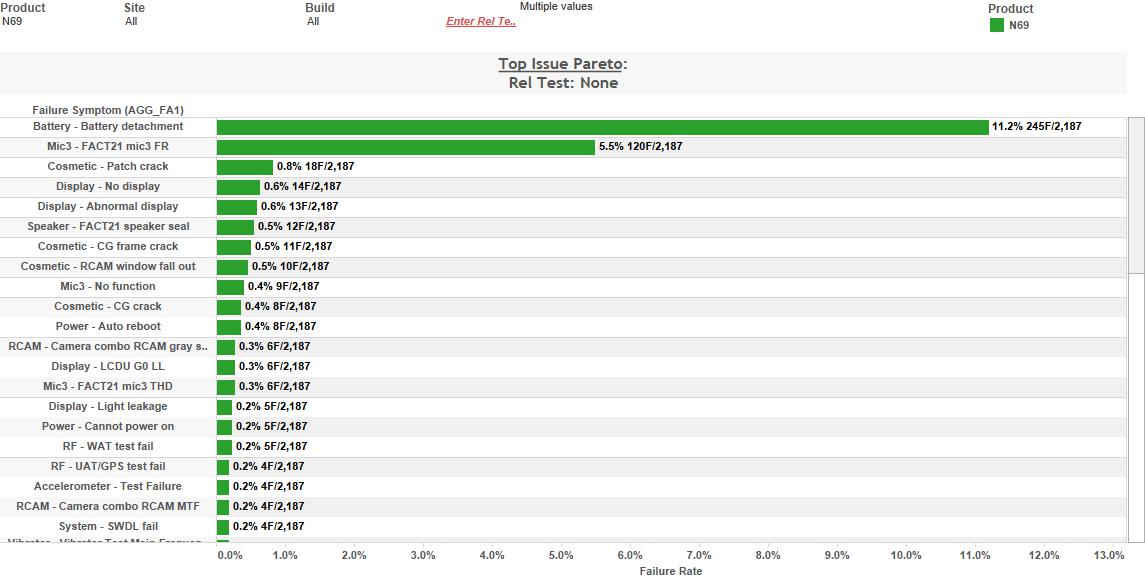
* 1. ORT Project; Sensor Line



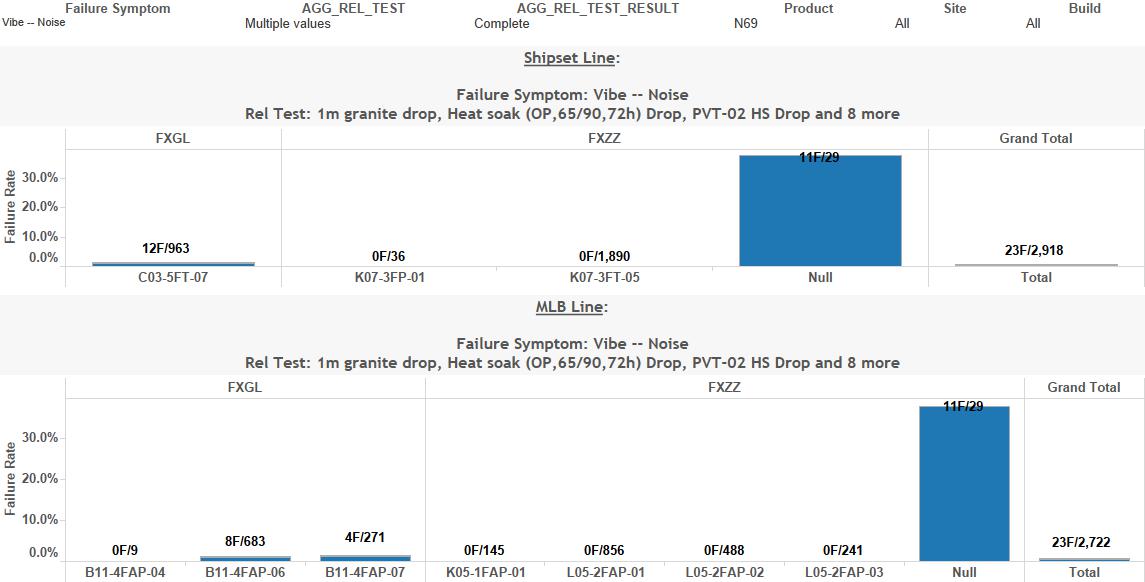
* 1. Line breakdown



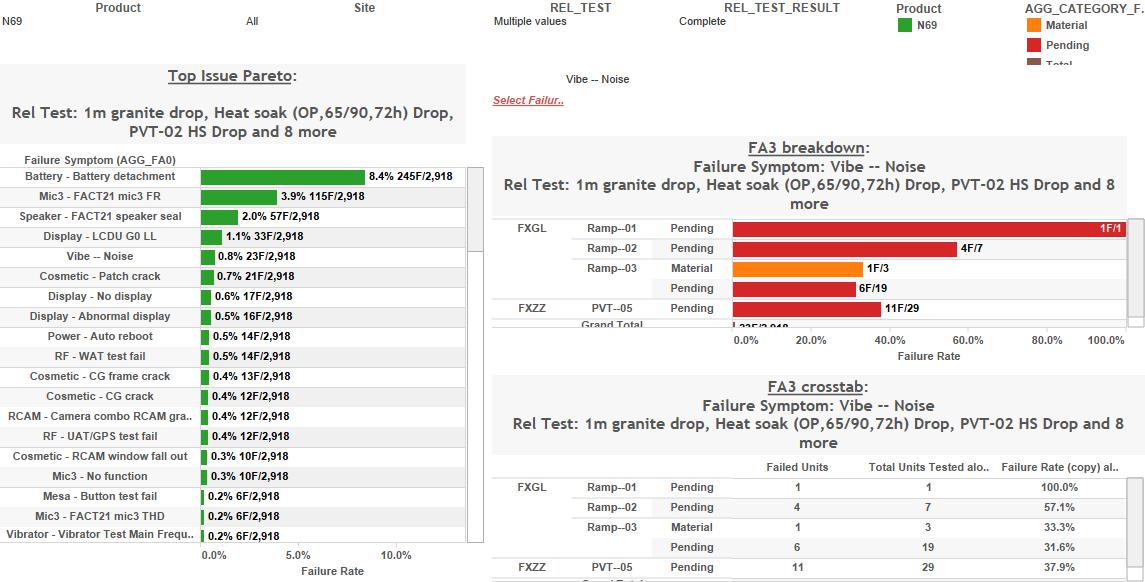
* 1. Reltest Combination, Pareto Chart



* 1. Shipset Line Breakdown



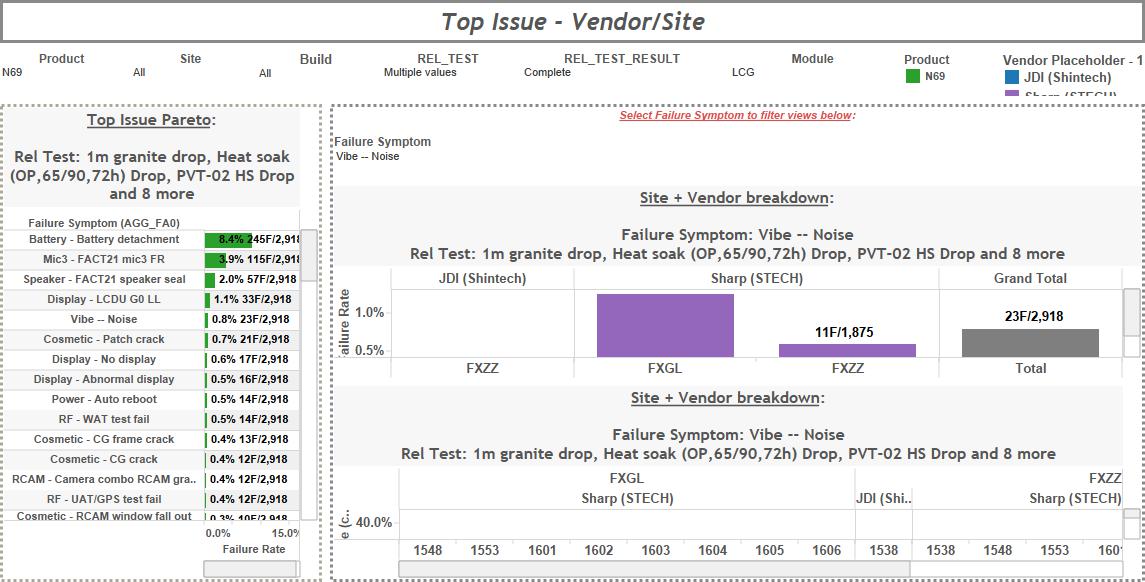
* 1. Top Issue



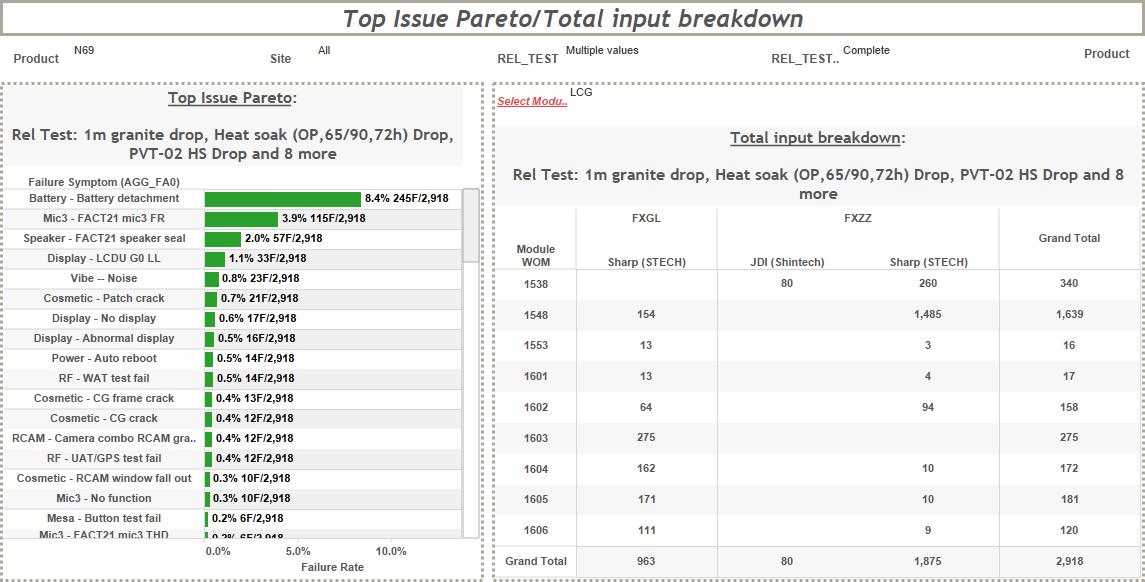
* 1. Top issue Summary



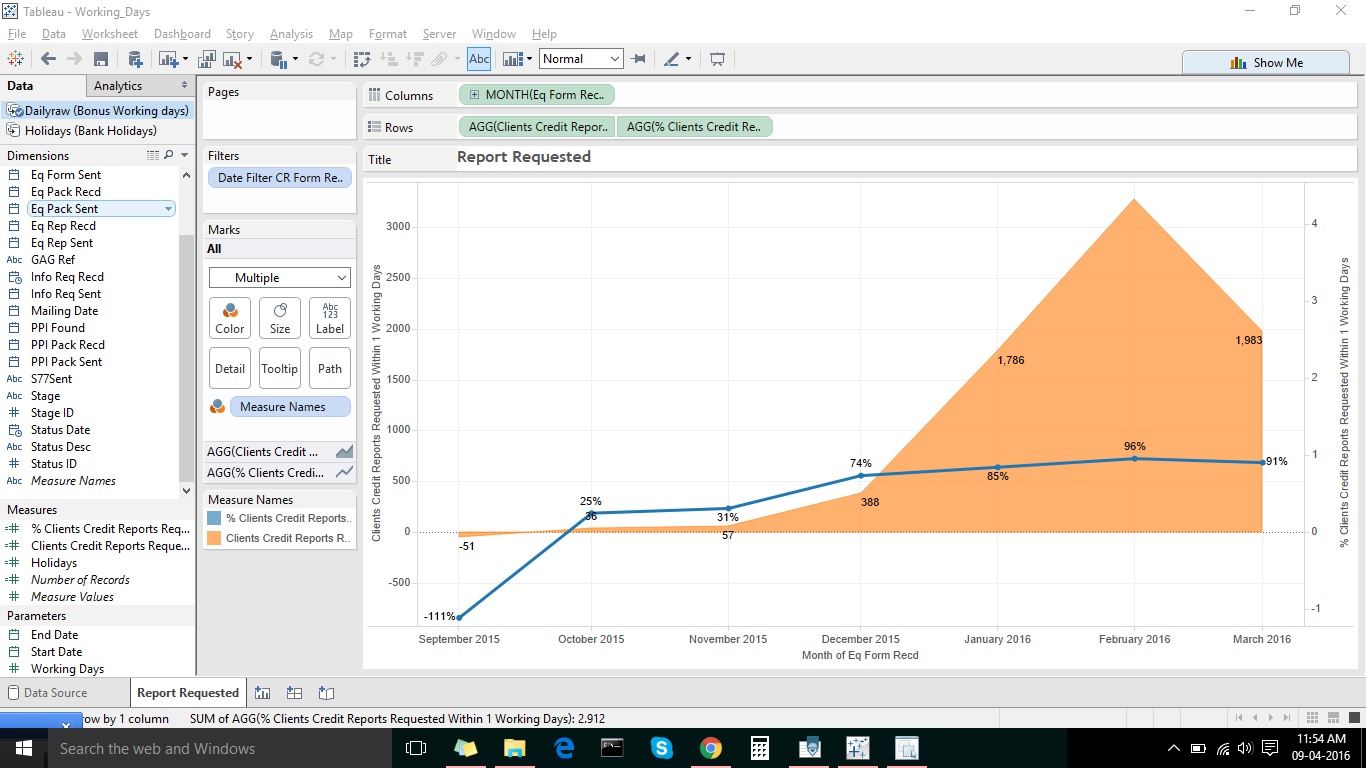
* 1. Top issue –Vendor



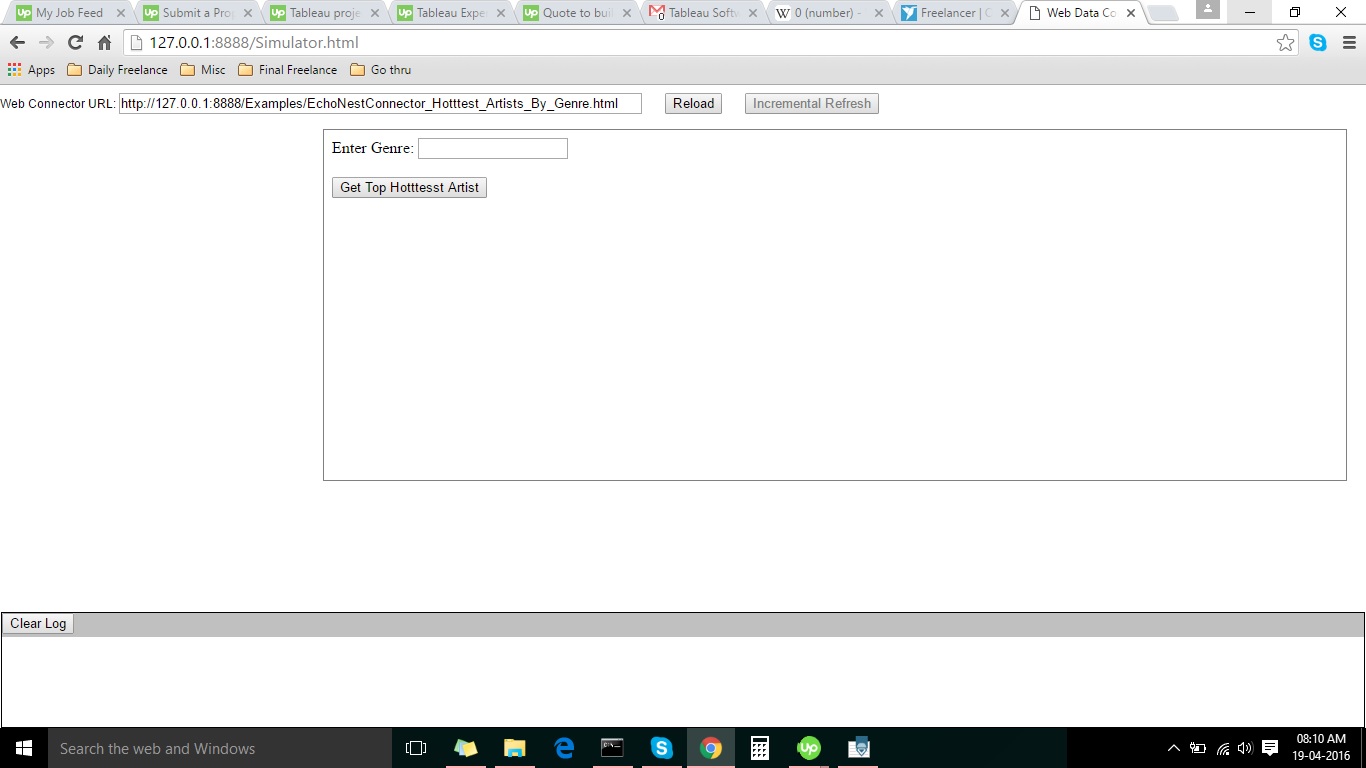
* 1. Total input breakdown



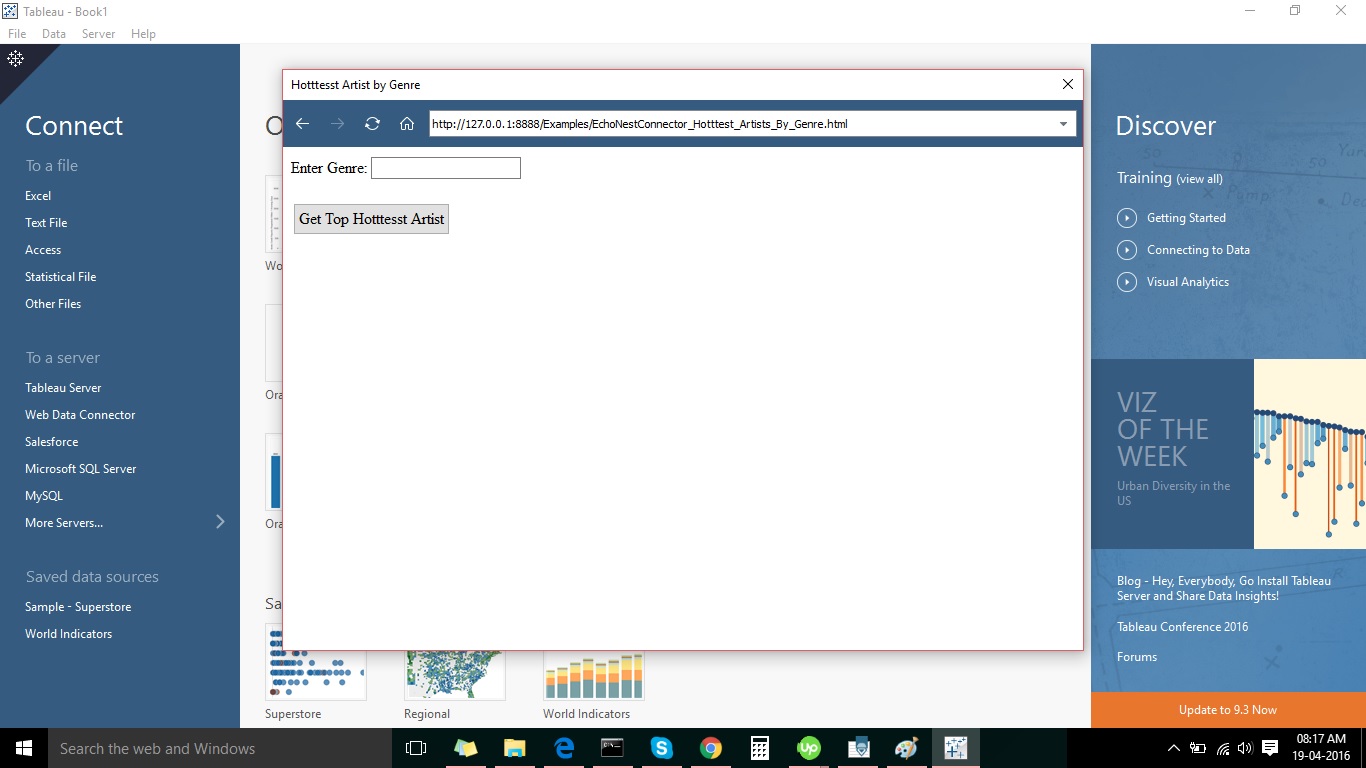
1. Holiday Project; Filter Holidays in report



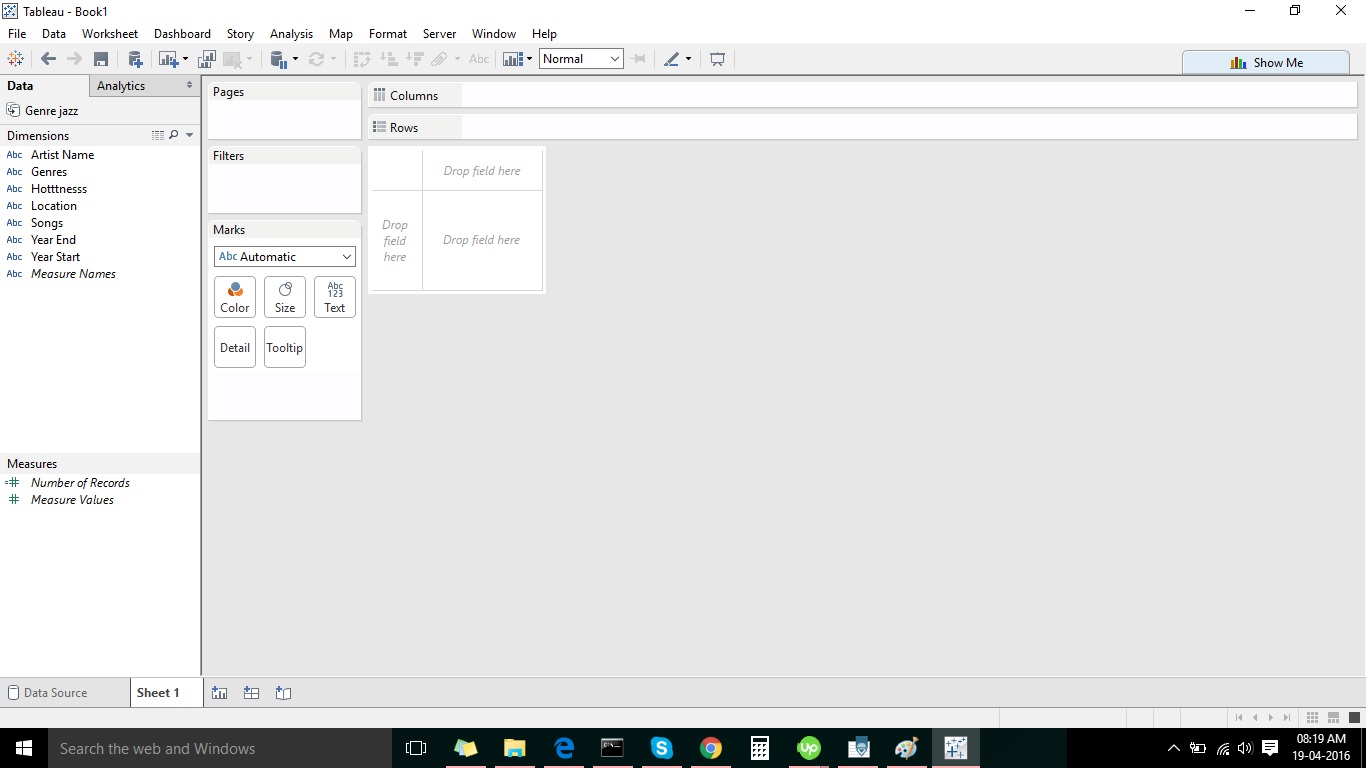
1. Echo Nest WDC Project; WDC, Python Web Server, API Call, JSON



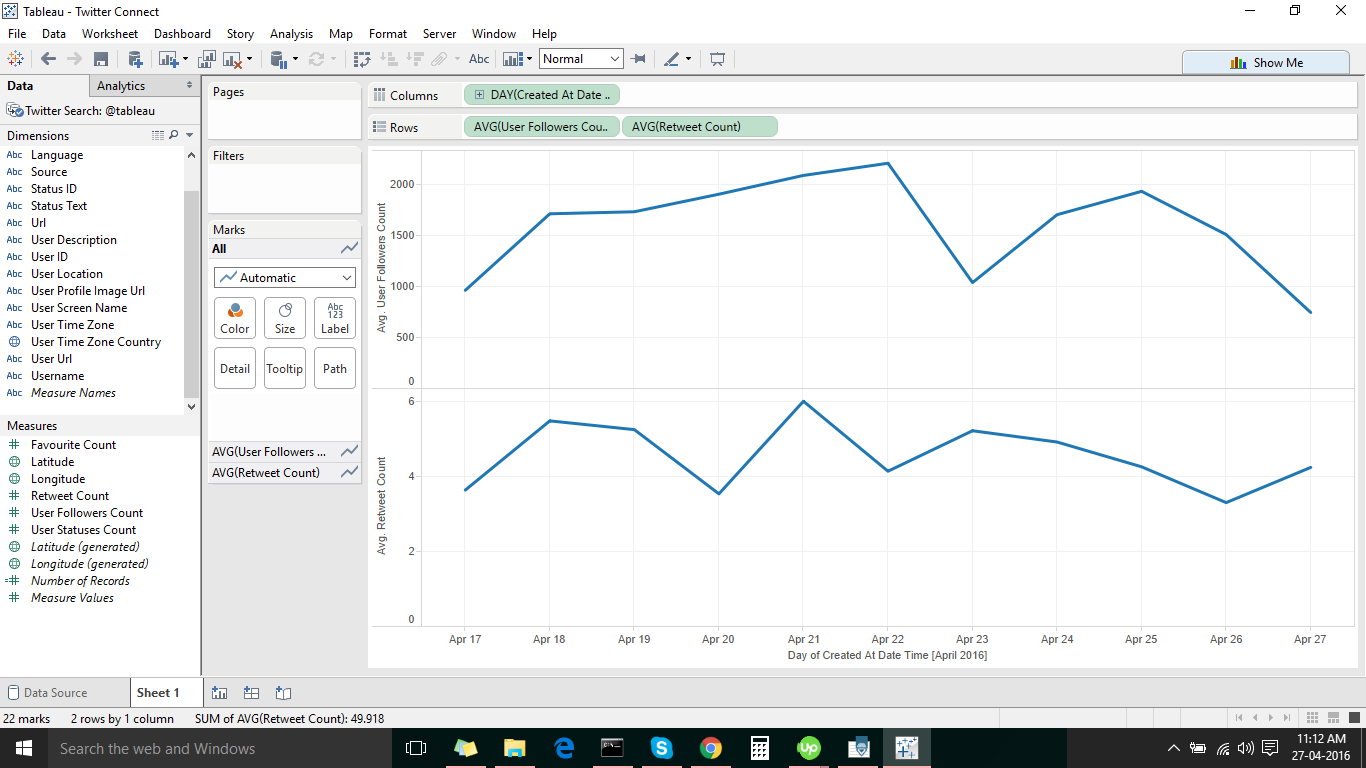
* 1. Connection at Tableau



* 1. Tableau Report



1. Twitter WDC Project; WDC, Python Web Server, API Call, JSON, Twitter Public & Private API



1. Salesforce Data Connect(SFDC) Project; Heroku, Sparkle, Tableau Online, Salesforce Canvas

