Sami Ali

Applied Statistics

Error Analysis of in situ Sea Surface Temperature Data

Background

Sea surface temperature (SST) is a basic measure of ocean surface conditions. SST measurements are a primary data input into applications such as climate research and modeling, weather forecasting including storm prediction and tracking, phenomena that affects weather and fisheries, and biological research on species and ecosystem health and productivity. SST can be measured by using in situ instruments, such as ships, oceanographic buoys, or by remote sensing, using sensors deployed on satellites. buoys can be drifting(DB) which freely travel by currents throughout the ocean, or moored(MB) which is fixed at a specific location in the ocean. In situ measurements are known to be affected by random error and by a variety of systematic biases, which depend on the measurement technique and the state of the instruments. The drawback to satellite SST is that it is an indirect. Thus, it is potentially more subject to error than direct, local measurements made by in situ instruments.

First, we load the python packages which we will need.

In [1]:

```
import pandas as pd
import numpy as np
import cartopy.crs as ccrs
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set(style="ticks")
```

The data does not have labels for columns so we create a list of labels.

In [2]:

```
col = ['YR','MO','DY','HR','LAT','LON','ISST','OSST','OERR','SI','ICflag','DS','VS','WDIR','WSPD','
SLP','AT','WBT','DPT','CLT','CLL']
```

Reading ship, DB and MB data with giving them the columns labels.

In [3]:

```
db = pd.read_csv('DB_32622.csv',names=col)
mb = pd.read_csv('MB_44043.csv',names=col)
sh = pd.read_csv('SH_FZCE.csv',names=col)
```

Checking if DB data is rightly loaded.

In [4]:

```
db.head()
```

Out[4]:

	YR	MO	DY	HR	LAT	LON	ISST	OSST	OERR	SI	...	DS	VS	WDIR	WSPD	SLP	AT	WBT	DPT	CLT	CLL
0	1997	8	25	12.80	1.13	246.61	26.4	26.96	0.35	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	1997	8	25	14.48	1.16	246.57	26.3	26.94	0.35	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	1997	8	25	20.28	1.16	246.57	26.6	26.94	0.35	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	1997	8	25	22.07	1.16	246.57	26.7	26.94	0.35	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	1997	8	26	1.48	1.29	246.33	26.6	26.82	0.35	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 21 columns

Summary for DB data to take overall look such how many entries we have, some idea about missing data and type of the entries since we expect all entries are numerical.

In [5]:

```
db.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20489 entries, 0 to 20488
Data columns (total 21 columns):
YR        20489 non-null int64
MO        20489 non-null int64
DY        20489 non-null int64
HR        20489 non-null float64
LAT       20489 non-null float64
LON       20489 non-null float64
ISST      20489 non-null float64
OSST      20489 non-null float64
OERR      20489 non-null float64
SI        0 non-null float64
ICflag    20489 non-null int64
DS        0 non-null float64
VS        0 non-null float64
WDIR      0 non-null float64
WSPD      0 non-null float64
SLP       0 non-null float64
AT        0 non-null float64
WBT       0 non-null float64
DPT       0 non-null float64
CLT       0 non-null float64
CLL       0 non-null float64
dtypes: float64(17), int64(4)
memory usage: 3.3 MB
```

From result we can see that there is no missing data in LAT, LON, ISST, OSST and ICflag which is perfect for analysis and all of them are numerical so the data is ready to use.

Statistics summary for each variable in DB data such mean, median, standard deviation, minimum and maximum.

In [6]:

```
db.describe()
```

Out[6]:

	YR	MO	DY	HR	LAT	LON	ISST	OSST	OERI
count	20489.000000	20489.000000	20489.000000	20489.000000	20489.000000	20489.000000	20489.000000	20489.000000	20489.00000
mean	2004.426326	6.753282	15.499048	12.042909	-9.733770	260.670695	25.614510	25.457426	0.21662
std	3.352436	3.226557	8.845705	6.742519	14.427877	13.374798	2.905259	3.056964	0.08344
min	1997.000000	1.000000	1.000000	0.000000	-31.350000	237.940000	18.100000	18.260000	0.10000
25%	2001.000000	4.000000	8.000000	6.730000	-21.140000	248.920000	24.000000	23.860000	0.14000
50%	2005.000000	7.000000	15.000000	11.880000	-16.050000	259.880000	26.400000	26.050000	0.22000
75%	2007.000000	9.000000	23.000000	16.980000	5.320000	273.660000	28.000000	28.140000	0.26000
max	2009.000000	12.000000	31.000000	23.980000	9.340000	282.460000	32.300000	30.970000	0.56000

8 rows × 21 columns

Checking if MB data is rightly loaded.

In [7]:

```
mb.head()
```

Out[7]:

	YR	MO	DY	HR	LAT	LON	ISST	OSST	OERR	SI	...	DS	VS	WDIR	WSPD	SLP	AT	WBT	DPT	CLT	CLL
0	2008	11	18	16	39.2	283.6	10.2	10.45	1.57	1	...	NaN	NaN	360.0	8.0	NaN	2.9	NaN	NaN	NaN	NaN
1	2008	11	18	17	39.2	283.6	10.2	10.45	1.57	1	...	NaN	NaN	340.0	9.0	NaN	3.0	NaN	NaN	NaN	NaN
2	2008	11	18	18	39.2	283.6	10.3	10.45	1.57	1	...	NaN	NaN	330.0	8.0	NaN	3.1	NaN	NaN	NaN	NaN
3	2008	11	18	19	39.2	283.6	10.3	10.45	1.57	1	...	NaN	NaN	340.0	10.0	NaN	3.3	NaN	NaN	NaN	NaN
4	2008	11	18	20	39.2	283.6	10.2	10.45	1.57	1	...	NaN	NaN	330.0	10.0	NaN	3.8	NaN	NaN	NaN	NaN

5 rows × 21 columns

Summary for MB data to take overall look such how many entries we have, some idea about missing data and type of the entries since we expect all entries are numerical.

In [8]:

```
mb.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9768 entries, 0 to 9767
Data columns (total 21 columns):
YR        9768 non-null int64
MO        9768 non-null int64
DY        9768 non-null int64
HR        9768 non-null int64
LAT       9768 non-null float64
LON       9768 non-null float64
ISST      9768 non-null float64
OSST      9768 non-null float64
OERR      9768 non-null float64
SI        9768 non-null int64
ICflag    9768 non-null int64
DS        0 non-null float64
VS        0 non-null float64
WDIR      8592 non-null float64
WSPD      9698 non-null float64
SLP       2709 non-null float64
AT        9717 non-null float64
WBT       0 non-null float64
DPT       0 non-null float64
CLT       0 non-null float64
CLL       0 non-null float64
dtypes: float64(15), int64(6)
memory usage: 1.6 MB
```

From result we can see that there is no missing data in LAT, LON, ISST, OSST and ICflag which is perfect for analysis and all of them are numerical so the data is ready to use.

Statistics summary for each variable in MB data such mean, median, standard deviation, minimum and maximum.

In [9]:

```
mb.describe()
```

Out[9]:

	YR	MO	DY	HR	LAT	LON	ISST	OSST	OERR	S
count	9768.000000	9768.000000	9768.000000	9768.000000	9.768000e+03	9.768000e+03	9768.000000	9768.000000	9768.000000	9768.
mean	2009.620495	8.225123	16.150799	11.495086	3.920000e+01	2.836000e+02	15.861538	15.012740	1.329059	1.
std	0.656085	3.156671	8.659130	6.941106	6.949464e-12	5.161647e-11	8.694649	8.175903	0.434740	0.
min	2008.000000	1.000000	1.000000	0.000000	3.920000e+01	2.836000e+02	-0.700000	-0.050000	0.520000	1.
25%	2009.000000	6.000000	9.000000	5.000000	3.920000e+01	2.836000e+02	8.800000	7.510000	0.930000	1.
50%	2010.000000	9.000000	17.000000	11.000000	3.920000e+01	2.836000e+02	15.800000	14.660000	1.380000	1.

	YR	MO	DY	HR	LAT	LON	ISST	OSST	OERR	
75%	2010.000000	11.000000	23.000000	18.000000	3.920000e+01	2.836000e+02	23.500000	22.260000	1.610000	1.
max	2010.000000	12.000000	31.000000	23.000000	3.920000e+01	2.836000e+02	32.000000	27.980000	2.080000	1.

8 rows × 21 columns

Checking if ship data is rightly loaded.

In [10]:

```
sh.head()
```

Out[10]:

	YR	MO	DY	HR	LAT	LON	ISST	OSST	OERR	SI	...	DS	VS	WDIR	WSPD	SLP	AT	WBT	DPT	CLT	CLL
0	2005	9	22	21	53.9	8.5	17.3	16.33	0.51	3	...	9.0	NaN	160.0	5.7	1022.2	16.0	NaN	9.4	NaN	NaN
1	2005	9	23	0	53.9	7.3	17.5	16.96	0.39	3	...	9.0	NaN	160.0	7.7	1019.8	16.4	NaN	10.7	NaN	NaN
2	2005	9	23	3	53.7	6.0	17.8	17.27	0.33	3	...	9.0	NaN	180.0	6.7	1017.5	15.8	NaN	11.2	NaN	NaN
3	2005	9	23	6	53.4	4.7	17.9	17.46	0.23	3	...	9.0	NaN	170.0	8.2	1015.3	16.3	NaN	13.2	NaN	NaN
4	2005	9	23	9	52.9	4.0	17.9	17.47	0.26	3	...	9.0	NaN	180.0	10.3	1014.1	17.3	NaN	13.8	NaN	NaN

5 rows × 21 columns

Summary for ship data to take overall look such how many entries we have, some idea about missing data and type of the entries since we expect all entries are numerical.

In [11]:

```
sh.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11010 entries, 0 to 11009
Data columns (total 21 columns):
YR        11010 non-null int64
MO        11010 non-null int64
DY        11010 non-null int64
HR        11010 non-null int64
LAT       11010 non-null float64
LON       11010 non-null float64
ISST      11010 non-null float64
OSST      10216 non-null float64
OERR      10216 non-null float64
SI        11010 non-null int64
ICflag    11010 non-null int64
DS        11007 non-null float64
VS        10951 non-null float64
WDIR      10785 non-null float64
WSPD      10785 non-null float64
SLP       11004 non-null float64
AT        10988 non-null float64
WBT       0 non-null float64
DPT       10988 non-null float64
CLT       423 non-null float64
CLL       408 non-null float64
dtypes: float64(15), int64(6)
memory usage: 1.8 MB
```

From result we can see also that there is no missing data in LAT, LON, ISST, OSST and ICflag which is perfect for analysis and all of them are numerical so the data is ready to use.

Statistics summary for each variable in ship data such mean, median, standard deviation, minimum and maximum.

In [12]:

```
sh.describe()
```

Out[12]:

Out[12]:

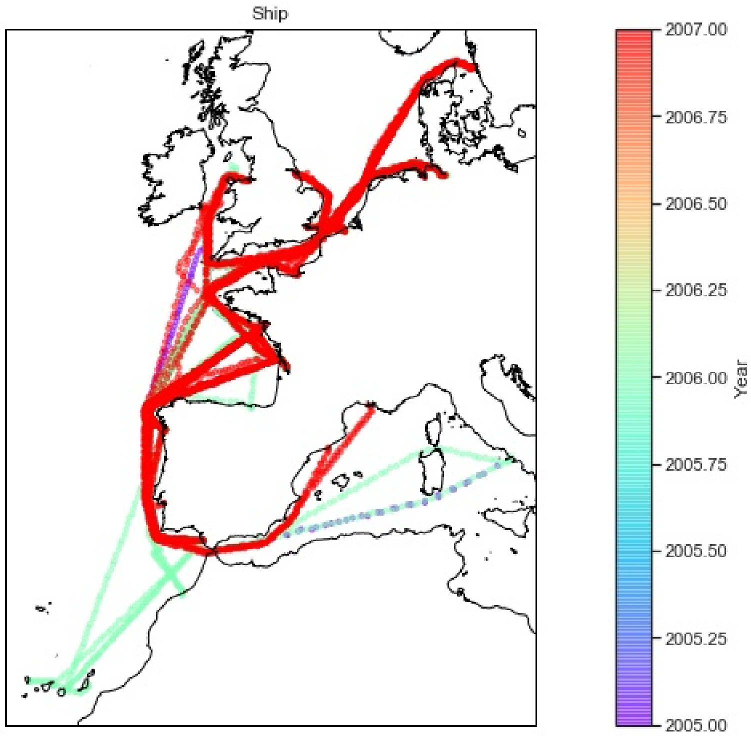
	YR	MO	DY	HR	LAT	LON	ISST	OSST	OERI
count	11010.000000	11010.000000	11010.000000	11010.000000	11010.000000	11010.000000	11010.000000	10216.000000	10216.00000
mean	2006.289282	5.160400	15.793551	11.475568	47.073115	248.709146	14.590082	13.557717	0.34401
std	0.625671	3.057251	8.685813	7.046721	5.756047	160.558029	5.391164	4.671999	0.15497
min	2005.000000	1.000000	1.000000	0.000000	27.900000	0.000000	0.600000	0.220000	0.12000
25%	2006.000000	3.000000	8.000000	5.000000	44.400000	9.800000	10.600000	9.867500	0.22000
50%	2006.000000	5.000000	16.000000	12.000000	47.000000	352.300000	14.200000	13.190000	0.31000
75%	2007.000000	7.000000	23.000000	18.000000	52.100000	356.900000	18.400000	17.240000	0.42000
max	2007.000000	12.000000	31.000000	23.000000	57.800000	359.900000	29.800000	24.610000	0.98000

8 rows × 21 columns

Ship tracks in different years.

In [13]:

```
plt.figure(figsize=(18,8))

ax = plt.axes(projection=ccrs.Mercator());
ax.coastlines(resolution='10m');

plt.scatter(sh.LON, sh.LAT, c=sh.YR,s=10,alpha=0.5,cmap='rainbow', transform=ccrs.Geodetic());
plt.colorbar(orientation="vertical",label='Year');
plt.title('Ship');
```

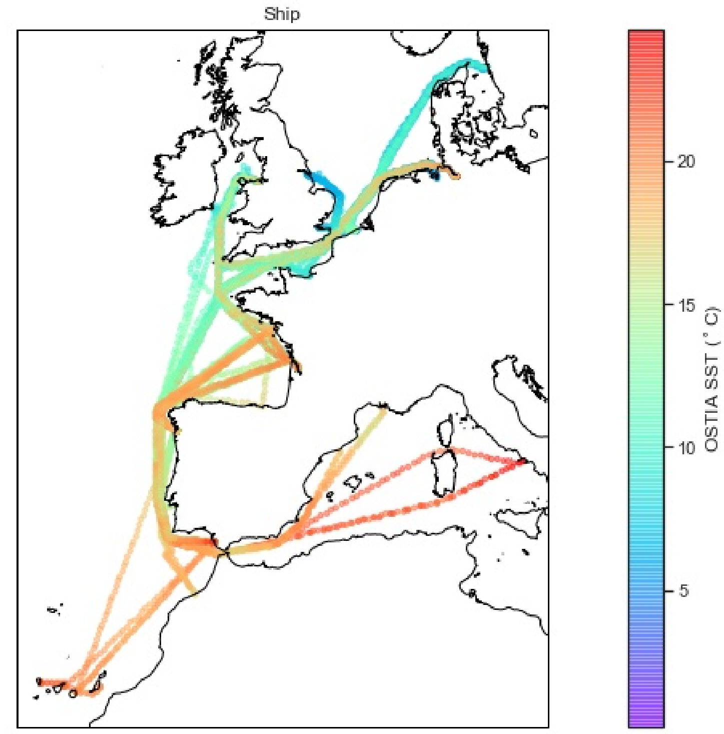


Our ship data has 3 years, but it looks like it has all data about 2006 but for others only some months.

SST on ship tracks.

In [14]:

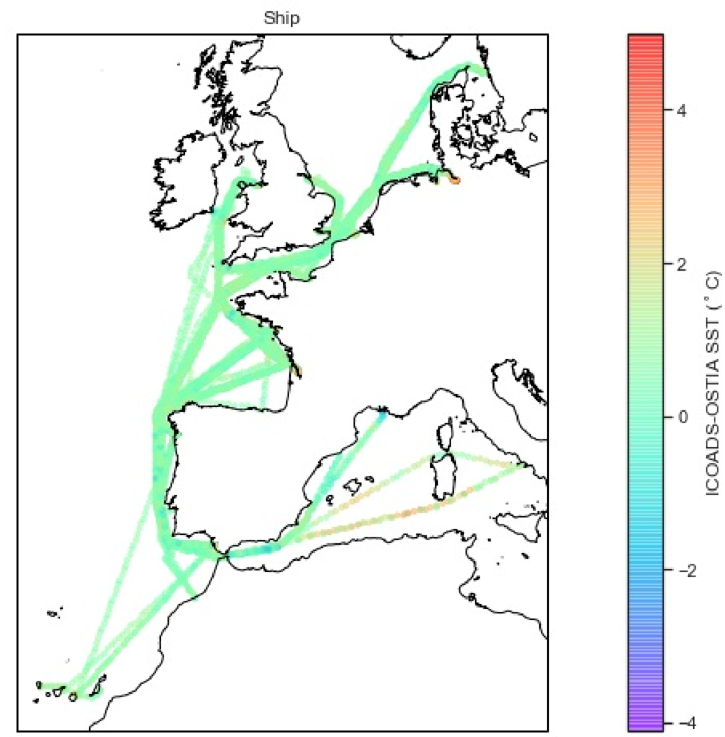
```
plt.figure(figsize=(18,8))

ax = plt.axes(projection=ccrs.Mercator())
ax.coastlines(resolution='10m')
plt.scatter(sh.LON, sh.LAT, c=sh.OSST,s=10,alpha=0.5,cmap='rainbow', transform=ccrs.Geodetic())
plt.colorbar(orientation="vertical",label='OSTIA SST ($^\circ$C)');
plt.title('Ship');
```



SST error on ship tracks. Only ICOADS SST values that passed its QC(only data file rows with ICflag= 1).

In [15]:

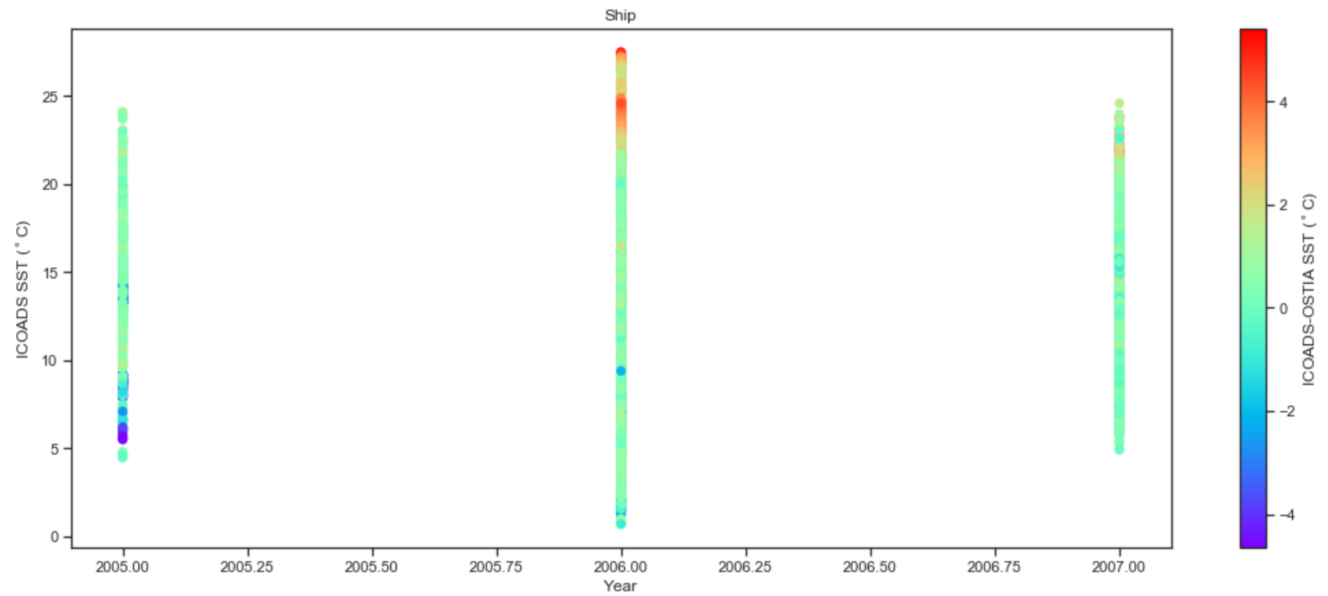
```
plt.figure(figsize=(18,8))

ax = plt.axes(projection=ccrs.Mercator())
ax.coastlines(resolution='10m')
plt.scatter(sh.LON[sh.ICflag==1], y=sh.LAT[sh.ICflag==1], c=sh.ISST[sh.ICflag==1] - sh.OSST[sh.ICfl
ag==1],s=10,alpha=0.5,
            cmap='rainbow', transform=ccrs.Geodetic())
plt.colorbar(orientation="vertical",label='ICOADS-OSTIA SST ($^\circ$C)');
plt.title('Ship');
```



SST observations at different times. The color shows ISST minus OSTIA differences

In [16]:

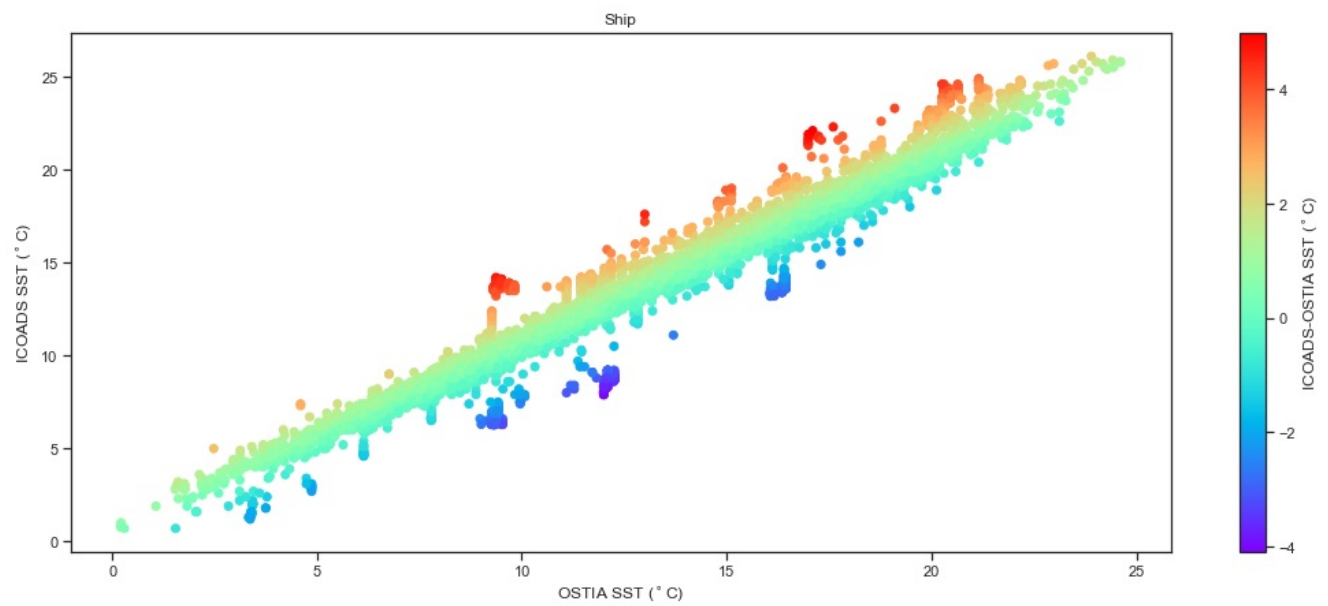
```
plt.figure(figsize=(18,7))

plt.scatter(x=sh.YR,y=sh.ISST,c=sh.ISST - sh.OSST,cmap='rainbow')
plt.colorbar(orientation="vertical",label='ICOADS-OSTIA SST ($^\circ$C)');
plt.xlabel('Year')
plt.ylabel('ICOADS SST ($^\circ$C)')
plt.title('Ship');
```



In situ observations vs satellite data analysis of SST. The color shows ISST minus OSTIA differences. Only ICOADS SST values that passed its QC(only data file rows with ICflag= 1).

In [17]:

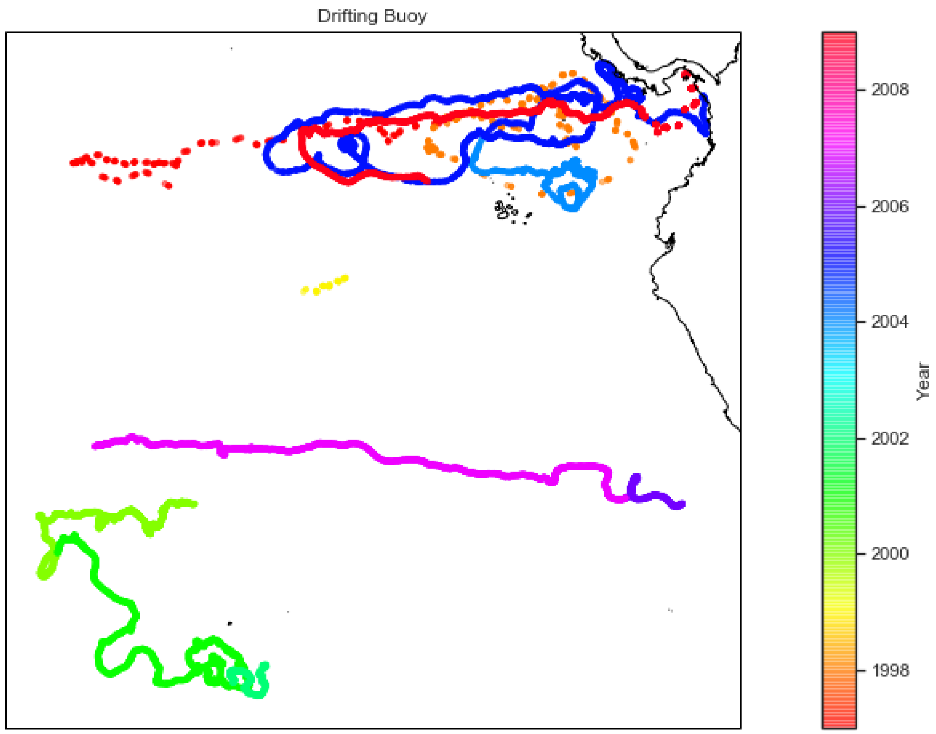
```
plt.figure(figsize=(18,7))

plt.scatter(x=sh.OSST[sh.ICflag==1],y=sh.ISST[sh.ICflag==1],c=sh.ISST[sh.ICflag==1] - sh.OSST[sh.IC
flag==1],cmap='rainbow')
plt.colorbar(orientation="vertical",label='ICOADS-OSTIA SST ($^\circ$C)');
plt.xlabel('OSTIA SST ($^\circ$C)')
plt.ylabel('ICOADS SST ($^\circ$C)')
plt.title('Ship');
```



Drifting buoy's trajectories in different years.

In [18]:

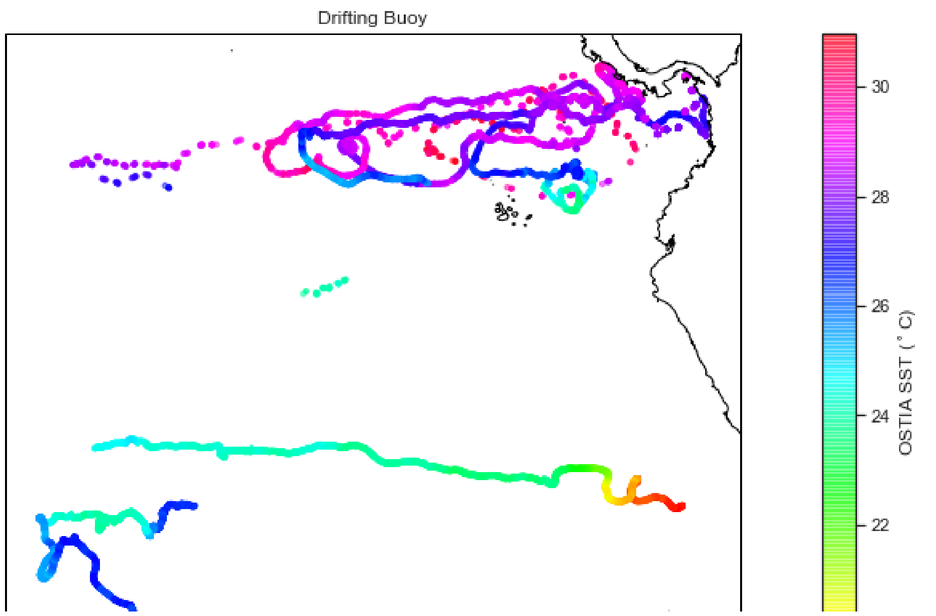
```
plt.figure(figsize=(18,8))

ax = plt.axes(projection=ccrs.Mercator())
ax.coastlines(resolution='10m')
plt.scatter(db.LON, db.LAT, c=db.YR,s=10,alpha=0.5,cmap='hsv', transform=ccrs.Geodetic())
plt.colorbar(orientation="vertical",label='Year');
plt.title('Drifting Buoy');
```



SST on DB trajectories.

In [19]:

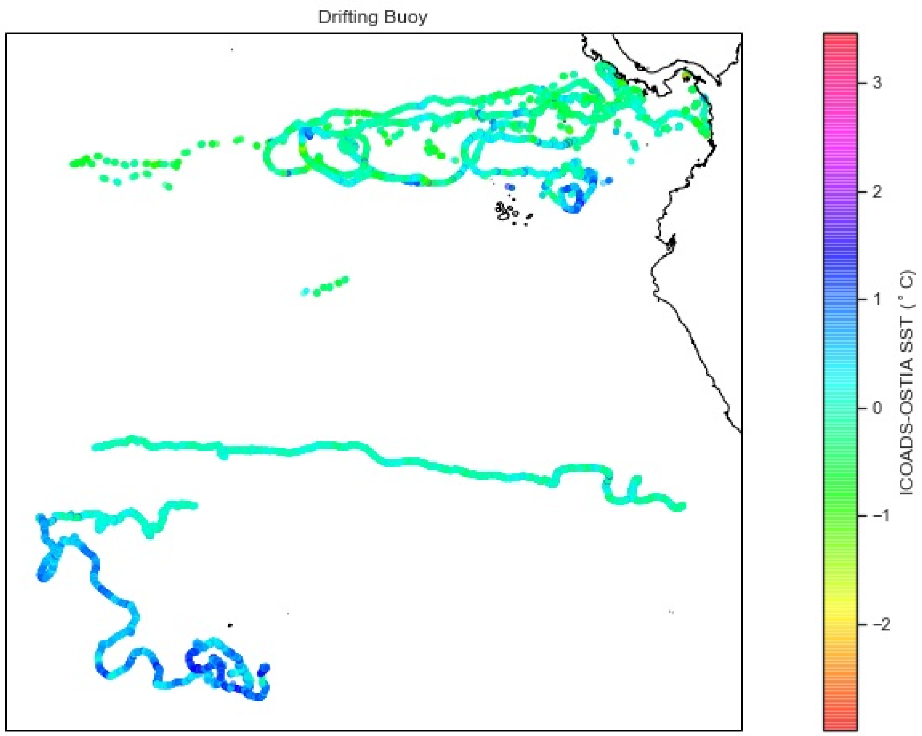
```
plt.figure(figsize=(18,8))

ax = plt.axes(projection=ccrs.Mercator())
ax.coastlines(resolution='10m')
plt.scatter(db.LON, db.LAT, c=db.OSST,s=10,alpha=0.5,cmap='hsv', transform=ccrs.Geodetic())
plt.colorbar(orientation="vertical",label='OSTIA SST ($^\circ$C)');
plt.title('Drifting Buoy');
```



SST error on DB trajectories. Only ICOADS SST values that passed its QC(only data file rows with ICflag= 1).

In [20]:

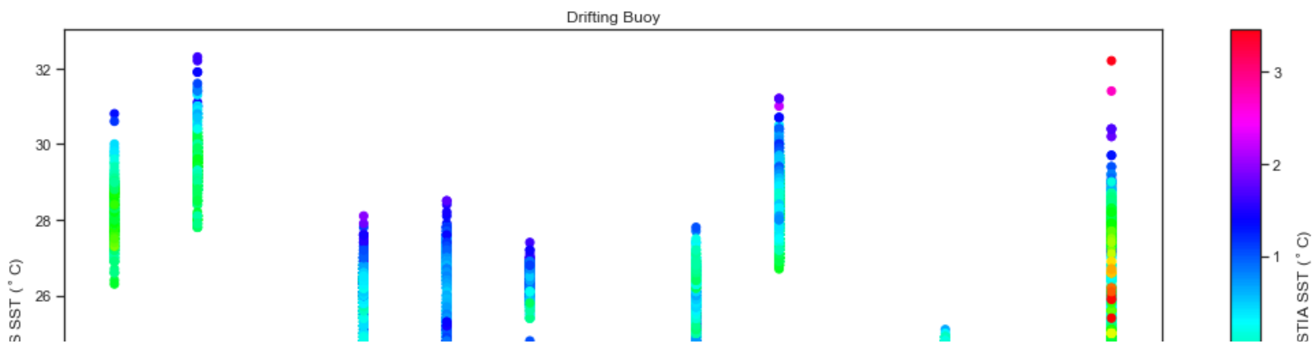
```
plt.figure(figsize=(18,8))

ax = plt.axes(projection=ccrs.Mercator())
ax.coastlines(resolution='10m')
plt.scatter(db.LON[db.ICflag==1], y=db.LAT[db.ICflag==1], c=db.ISST[db.ICflag==1] - db.OSST[db.ICfl
ag==1],s=10,alpha=0.5,cmap='hsv', transform=ccrs.Geodetic())
plt.colorbar(orientation="vertical",label='ICOADS-OSTIA SST ($^\circ$C)');
plt.title('Drifting Buoy');
```

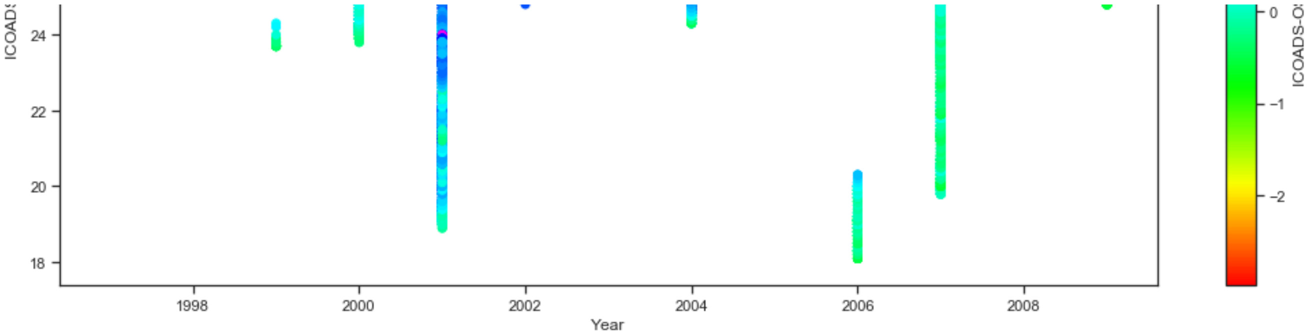


SST observations at different times.

In [21]:

```
plt.figure(figsize=(18,8))

plt.scatter(x=db.YR[db.ICflag==1],y=db.ISST[db.ICflag==1],c=db.ISST[db.ICflag==1] - db.OSST[db.ICfl
ag==1],cmap='hsv')
plt.colorbar(orientation="vertical",label='ICOADS-OSTIA SST ($^\circ$C)');
plt.xlabel('Year')
plt.ylabel('ICOADS SST ($^\circ$C)')
plt.title('Drifting Buoy');
```

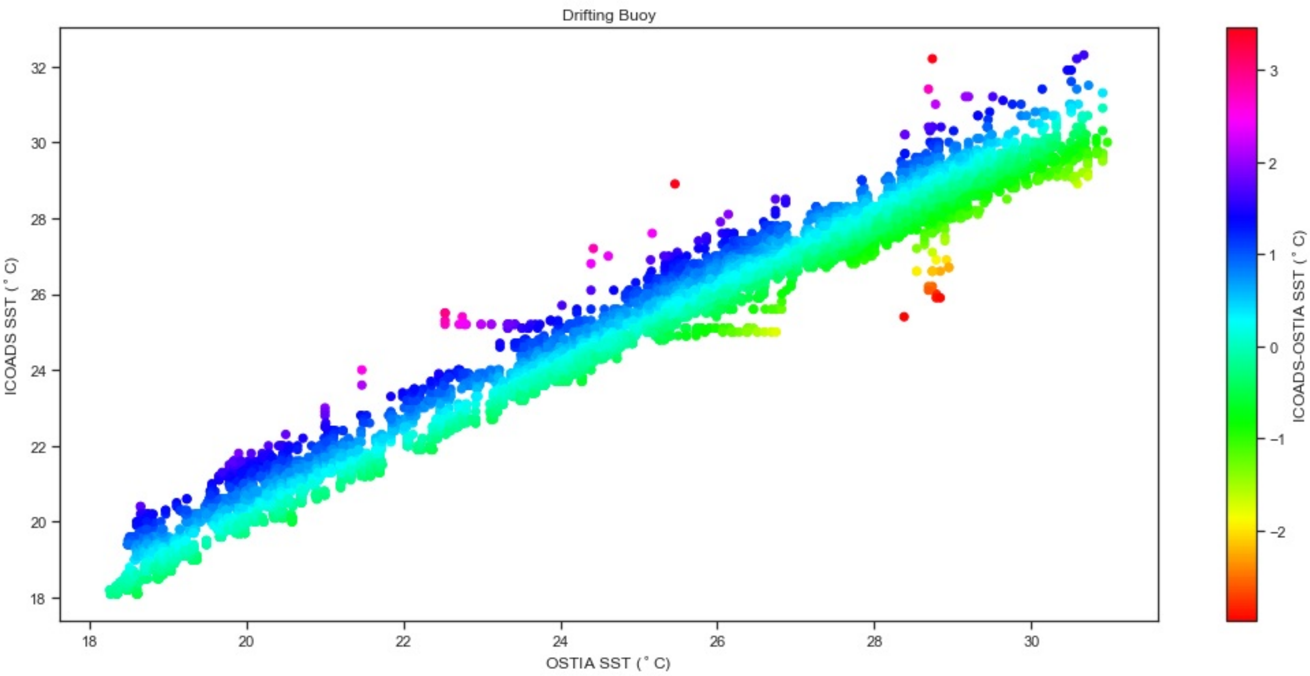




In situ observations vs satellite data analysis of SST DB.

In [22]:

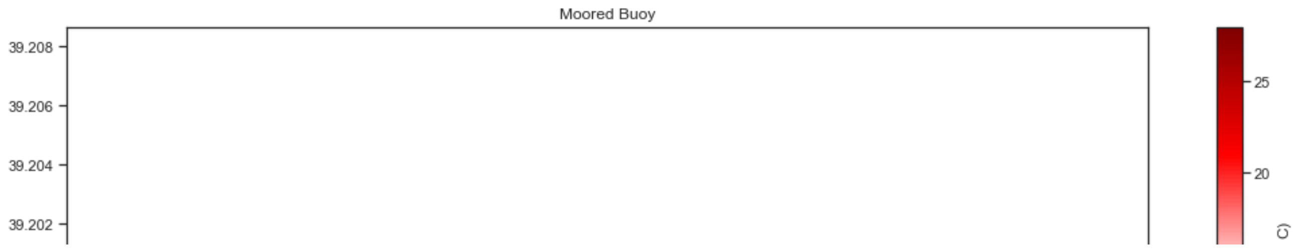
```
plt.figure(figsize=(18,8))
plt.scatter(x=db.OSST[db.ICflag==1],y=db.ISST[db.ICflag==1],c=db.ISST[db.ICflag==1] - db.OSST[db.IC
flag==1],cmap='hsv')
plt.colorbar(orientation="vertical",label='ICOADS-OSTIA SST ($^\circ$C)');
plt.xlabel('OSTIA SST ($^\circ$C)')
plt.ylabel('ICOADS SST ($^\circ$C)')
plt.title('Drifting Buoy');
```

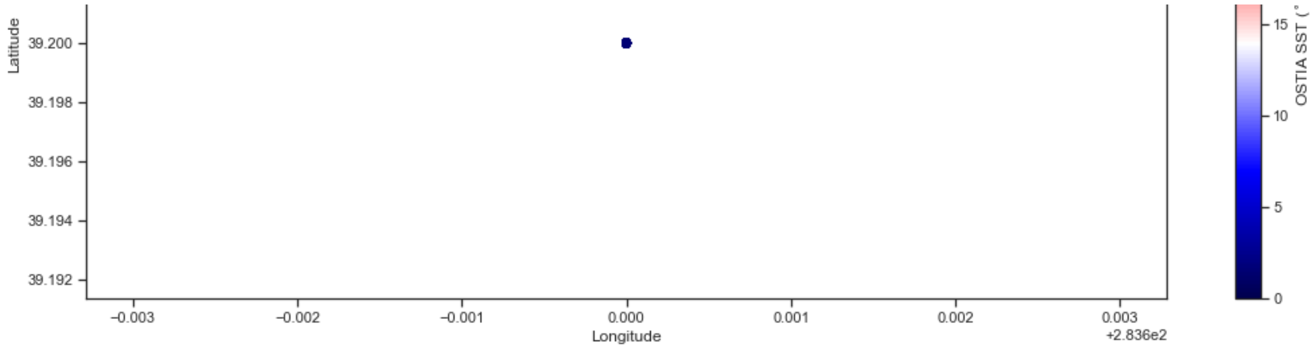


SST at an MB.

In [23]:

```
plt.figure(figsize=(18,7))

plt.scatter(x=mb.LON,y=mb.LAT,c=mb.OSST,cmap='seismic')
plt.colorbar(orientation="vertical",label='OSTIA SST ($^\circ$C)');
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Moored Buoy');
```

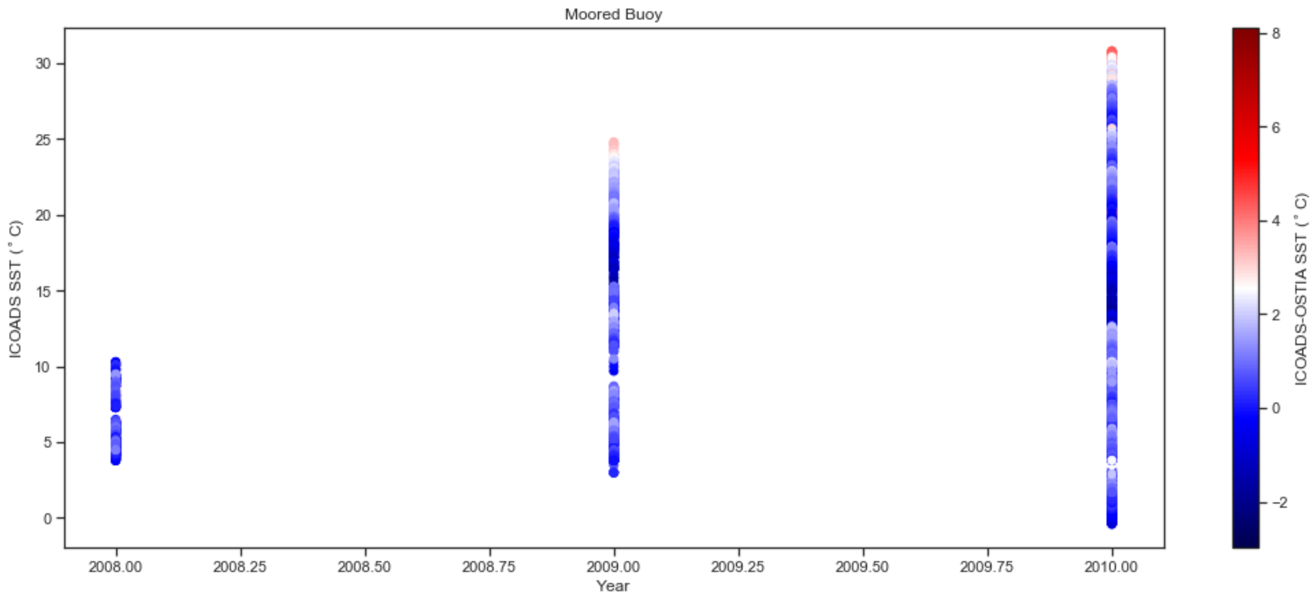




SST observations at MB at different times.

In [24]:

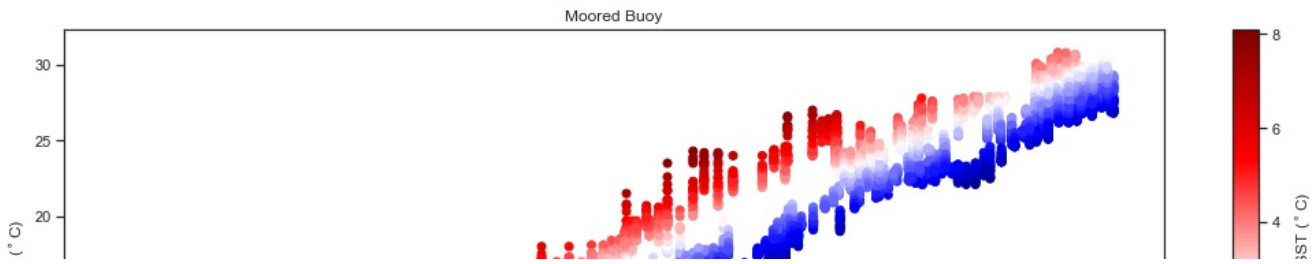
```
plt.figure(figsize=(18,7))

plt.scatter(x=mb.YR[mb.ICflag==1],y=mb.ISST[mb.ICflag==1],c=mb.ISST[mb.ICflag==1] - mb.OSST[mb.ICfl
ag==1],cmap='seismic')
plt.colorbar(orientation="vertical",label='ICOADS-OSTIA SST ($^\circ$C)');
plt.xlabel('Year')
plt.ylabel('ICOADS SST ($^\circ$C)')
plt.title('Moored Buoy');
```

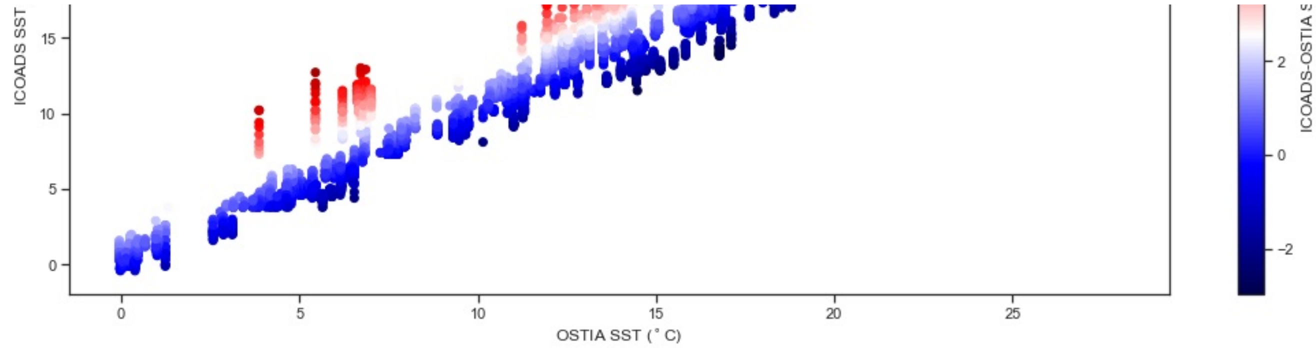


In situ observations vs satellite data analysis of SST.

In [25]:

```
plt.figure(figsize=(18,7))

plt.scatter(x=mb.OSST[mb.ICflag==1],y=mb.ISST[mb.ICflag==1],c=mb.ISST[mb.ICflag==1] - mb.OSST[mb.IC
flag==1],cmap='seismic')
plt.colorbar(orientation="vertical",label='ICOADS-OSTIA SST ($^\circ$C)');
plt.xlabel('OSTIA SST ($^\circ$C)')
plt.ylabel('ICOADS SST ($^\circ$C)')
plt.title('Moored Buoy');
```



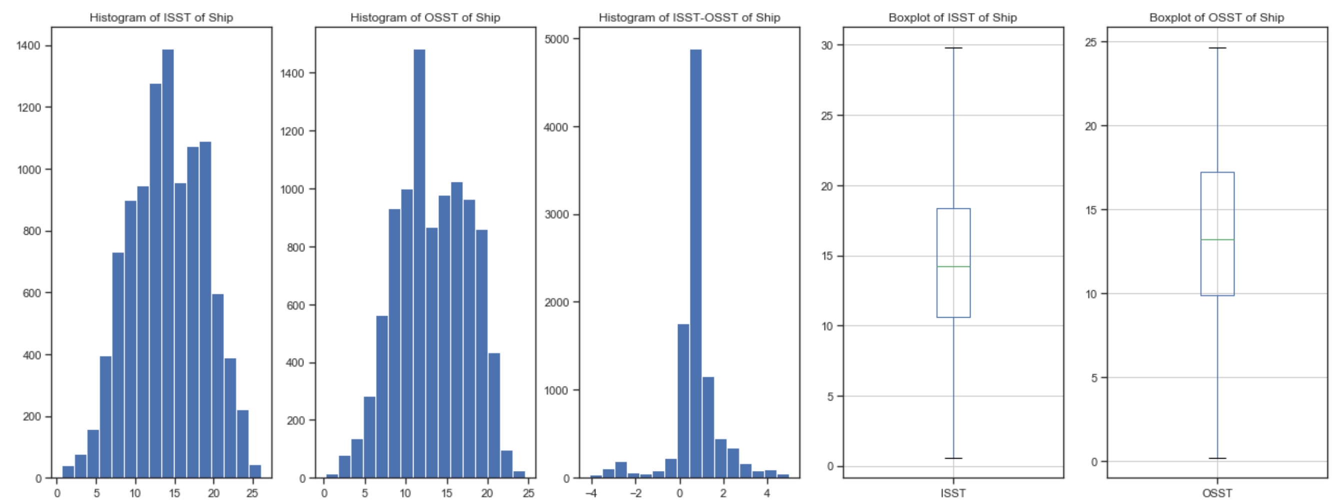


Investigating ISST, OSST and ISST-OSST distributions and outliers for only QC in the ship.

In [26]:

```
plt.figure(figsize=(22,8))

plt.subplot(1,5,1)
plt.hist(sh.ISST[sh.ICflag==1],bins=16);
plt.title('Histogram of ISST of Ship')

plt.subplot(1,5,2)
plt.hist(sh.OSST[sh.ICflag==1],bins=16);
plt.title('Histogram of OSST of Ship')


plt.subplot(1,5,3)
plt.hist(sh.ISST[sh.ICflag==1] - sh.OSST[sh.ICflag==1],bins=16);
plt.title('Histogram of ISST-OSST of Ship');

plt.subplot(1,5,4)
sh.boxplot(column=['ISST'])
plt.title('Boxplot of ISST of Ship');

plt.subplot(1,5,5)
sh.boxplot(column=['OSST'])
plt.title('Boxplot of OSST of Ship');
```

```
C:\Users\admin1\Anaconda3\lib\site-packages\numpy\lib\histograms.py:839: RuntimeWarning: invalid v
alue encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\admin1\Anaconda3\lib\site-packages\numpy\lib\histograms.py:840: RuntimeWarning: invalid v
alue encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

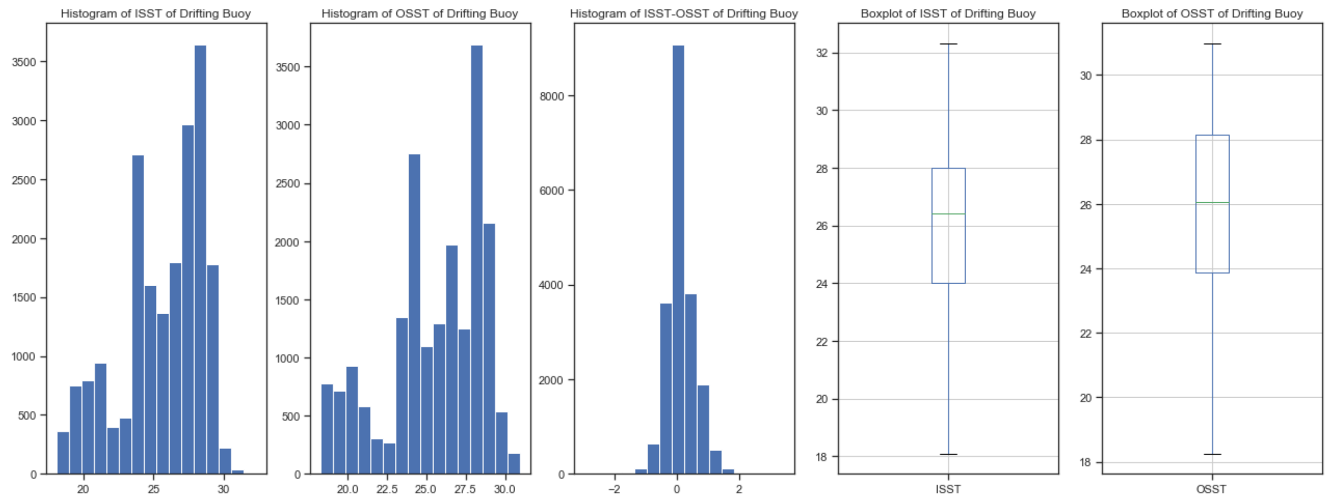


From histograms and boxplots above as we expect the distribution of ISST and OSST are not normal. The distribution of ISST-OSST is also not normal. We will test these observations by doing normality test later. We notice also both of ISST and OSST do not have outliers which minimize our analysis error. We can see that many difference(ISST-OSST) points are between 0 and 1 but some of them are aound 4 so the error between in situ and satelite is probaby not big. We will confirm later if their means are significantly different or not

Investigating ISST, OSST and ISST-OSST distributions and outliers for only QC in the DB.

In [27]:

```
plt.figure(figsize=(22,8))

plt.subplot(1,5,1)
plt.hist(db.ISST[db.ICflag==1],bins=16);
plt.title('Histogram of ISST of Drifting Buoy')

plt.subplot(1,5,2)
plt.hist(db.OSST[db.ICflag==1],bins=16);
plt.title('Histogram of OSST of Drifting Buoy')


plt.subplot(1,5,3)
plt.hist(db.ISST[db.ICflag==1] - db.OSST[db.ICflag==1],bins=16);
plt.title('Histogram of ISST-OSST of Drifting Buoy');

plt.subplot(1,5,4)
db.boxplot(column=['ISST'])
plt.title('Boxplot of ISST of Drifting Buoy');

plt.subplot(1,5,5)
db.boxplot(column=['OSST'])
plt.title('Boxplot of OSST of Drifting Buoy');
```



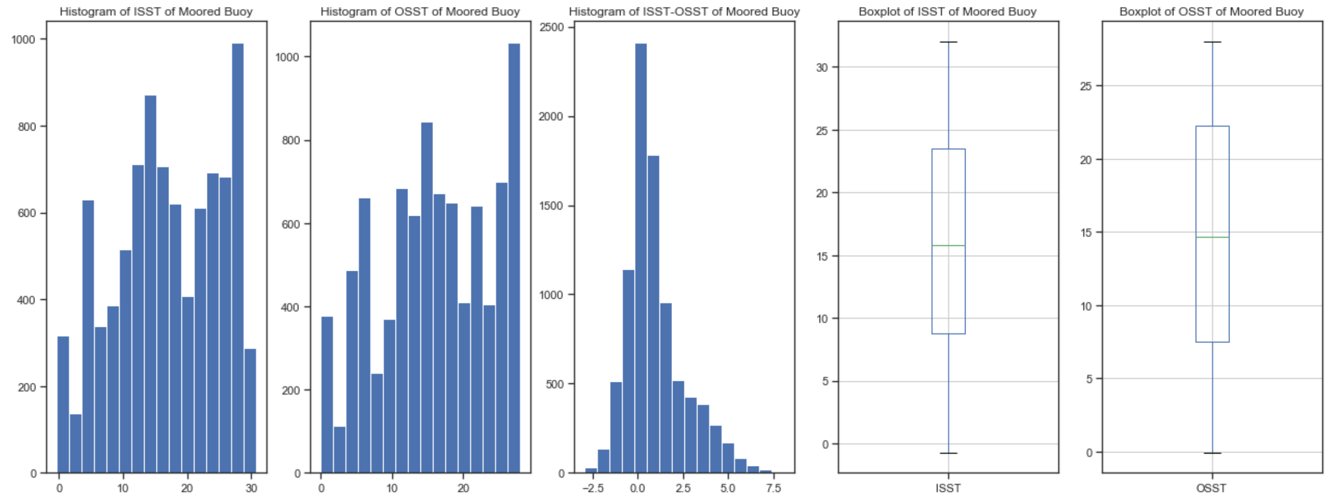
From histograms and boxplots above as we expect the distribution of ISST and OSST are not normal. The distribution of ISST-OSST looks like near to be normal. We will test these observations by doing normality test later. We notice also both of ISST and OSST do not have any outliers which reduce the probability of bias. We can see that the maximum difference(ISST-OSST) is less than 2 so the error between in situ and satelite is mostly not big. We will confirm later if their means are significantly different or not.

Investigating ISST, OSST and ISST-OSST distributions and outliers for QC in the MB.

In [28]:

```
plt.figure(figsize=(22,8))

plt.subplot(1,5,1)
plt.hist(mb.ISST[mb.ICflag==1],bins=16);
plt.title('Histogram of ISST of Moored Buoy')

plt.subplot(1,5,2)
plt.hist(mb.OSST[mb.ICflag==1],bins=16);
plt.title('Histogram of OSST of Moored Buoy')


plt.subplot(1,5,3)
plt.hist(mb.ISST[mb.ICflag==1] - mb.OSST[mb.ICflag==1],bins=16);
plt.title('Histogram of ISST-OSST of Moored Buoy');

plt.subplot(1,5,4)
mb.boxplot(column=['ISST'])
```

```
plt.title('Boxplot of ISST of Moored Buoy');

plt.subplot(1,5,5)
mb.boxplot(column=['OSST'])
plt.title('Boxplot of OSST of Moored Buoy');
```



From histograms and boxplots above as we expect the distribution of ISST and OSST are not normal. The distribution of ISST-OSST is right skewed(not normal). We will test these observations by doing normality test later. We notice also both of ISST and OSST do not have any outliers which help us in our statistics analysis. We can see that many difference(ISST-OSST) points are between 0 and 1 but the maximum point is around 7 so the error between in situ and satelite can be big. Thus, we expect their means are significantly different.

Hypothesis Testing

Normality Tests

H0: the data has a Normal distribution.

H1: the data does not have a Normal distribution.

For ship data:

In [30]:

```
# Example of the Shapiro-Wilk Normality Test
from scipy.stats import shapiro

data1 = sh.ISST[sh.ICflag==1]
stat1, p1 = shapiro(data1[:1000])
print('P value for normality test of ISST = ',p1)
if p1 > 0.05:
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of ISST is probably normal.')
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so ISST probably\ndoes not have a normal distribution as we expected.')

data2 = sh.OSST[sh.ICflag==1]
stat2, p2 = shapiro(data2[:1000])
print('\nP value for normality test of OSST = ',p2)
if p2 > 0.05:
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of OSST is probably normal.\nIt is not as we expected but it can happen by a chance.'
)
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so OSST probably does not have a normal distribution as we expected.')

ddd = sh.ISST[sh.ICflag==1] - sh.OSST[sh.ICflag==1]
stat2, p222 = shapiro(ddd[:1000])
print('\nP value for normality test of ISST-OSST = ',p222)
if p222 > 0.05:
```

```
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of ISST-OSST is probably normal.')
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so ISST-OSST probably does not have a normal distribution as we expected.')
```

```
P value for normality test of ISST =  6.002080681355437e-07
Since P value is smaller than 0.05, we can reject the null hypothesis and support the alternative
hypothesis so ISST probably
does not have a normal distribution as we expected.

P value for normality test of OSST =  1.0
Since P value is bigger than 0.05, we can not reject the null hypothesis so the distribution of OS
ST is probably normal.
It is not as we expected but it can happen by a chance.

P value for normality test of ISST-OSST =  1.0
Since P value is bigger than 0.05, we can not reject the null hypothesis so the distribution of IS
ST-OSST is probably normal.
```

For drifting buoy data:

In [32]:

```
data1 = db.ISST[db.ICflag==1]
stat1, p1 = shapiro(data1[:1000])
print('P value for normality test of ISST = ',p1)
if p1 > 0.05:
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of ISST is probably normal.')
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so ISST probably\ndoes not have a normal distribution as we expected and also f
rom histograms.')

data2 = db.OSST[db.ICflag==1]
stat2, p2 = shapiro(data2[:1000])
print('\nP value for normality test of OSST = ',p2)
if p2 > 0.05:
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of OSST is probably normal.')
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so OSST probably \ndoes not have a normal distribution as we expected and also
from histograms.')

ddd = db.ISST[db.ICflag==1] - db.OSST[db.ICflag==1]
stat2, p222 = shapiro(ddd[:1000])
print('\nP value for normality test of ISST-OSST = ',p222)
if p222 > 0.05:
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of ISST-OSST is probably normal.')
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so ISST-OSST \nprobably does not have a normal distribution as we expected and
also from histograms.')
```

```
P value for normality test of ISST =  6.618197117793762e-20
Since P value is smaller than 0.05, we can reject the null hypothesis and support the alternative
hypothesis so ISST probably
does not have a normal distribution as we expected and also from histograms.

P value for normality test of OSST =  6.090351709382698e-24
Since P value is smaller than 0.05, we can reject the null hypothesis and support the alternative
hypothesis so OSST probably
does not have a normal distribution as we expected and also from histograms.

P value for normality test of ISST-OSST =  8.297354231641407e-13
Since P value is smaller than 0.05, we can reject the null hypothesis and support the alternative
hypothesis so ISST-OSST
probably does not have a normal distribution as we expected and also from histograms.
```

For moored buoy:

For moored buoy:

In [33]:

```
data1 = mb.ISST[mb.ICflag==1]
stat1, p1 = shapiro(data1[:1000])
print('P value for normality test of ISST = ',p1)
if p1 > 0.05:
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of ISST is probably normal.')
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so ISST probably\ndoes not have a normal distribution as we expected and also f
rom histograms.')

data2 = mb.OSST[mb.ICflag==1]
stat2, p2 = shapiro(data2[:1000])
print('\nP value for normality test of OSST = ',p2)
if p2 > 0.05:
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of OSST is probably normal.')
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so OSST probably \ndoes not have a normal distribution as we expected and also
from histograms.')

ddd = mb.ISST[mb.ICflag==1] - mb.OSST[mb.ICflag==1]
stat2, p222 = shapiro(ddd[:1000])
print('\nP value for normality test of ISST-OSST = ',p222)
if p222 > 0.05:
 print('Since P value is bigger than 0.05, we can not reject the null hypothesis so the
distribution of ISST-OSST is probably normal.')
else:
 print('Since P value is smaller than 0.05, we can reject the null hypothesis and support the alte
rnative hypothesis so ISST-OSST \nprobably does not have a normal distribution as we expected and
also from histograms.')
```

```
P value for normality test of ISST =  6.688103858064229e-38
Since P value is smaller than 0.05, we can reject the null hypothesis and support the alternative
hypothesis so ISST probably
does not have a normal distribution as we expected and also from histograms.

P value for normality test of OSST =  1.1109332795584 06e-37
Since P value is smaller than 0.05, we can reject the null hypothesis and support the alternative
hypothesis so OSST probably
does not have a normal distribution as we expected and also from histograms.

P value for normality test of ISST-OSST =  7.36877364682087e-11
Since P value is smaller than 0.05, we can reject the null hypothesis and support the alternative
hypothesis so ISST-OSST
probably does not have a normal distribution as we expected and also from histograms.
```

Now, we need to investigate the relation between ISST and OSST. We will study it first within each group and then between all of them. After doing normality test, we can not use Z-test since data is not normal so we will use T-test to test the means by each group. For testing the means between all groups, we will use ANOVA. Therefore, we will use the difference between ISST and OSST as a good indicator if the errors are the same with ship, DB and MB. Furthermore, we will do correlation test to check independency between ISST and OSST.

Student’s t-test

H0: the means of ISST and OSST are equal.

H1: the means of ISST and OSST are unequal.

In [34]:

```
# Example of the Student's t-test
from scipy.stats import ttest_ind

data11 = sh.ISST[sh.ICflag==1]
data22 = sh.OSST[sh.ICflag==1]
stat12, p12 = ttest_ind(data11, data22,nan_policy='omit')
print('P value of t-test for ship = ',p12)
if p12 > 0.05:
```

```
if p12 > 0.05:
 print('We can not reject the null hypothesis since P value is bigger than 0.05 so probably ISST a
nd OSST have the same\ndistribution.')
else:
 print('We reject the null hypothesis and support the alternative one so ISST and OSST probably ha
ve different distributions\n(i.e., the means are unequal). This can be an evidence that the error
between them is significant.')

data33 = db.ISST[db.ICflag==1]
data44 = db.OSST[db.ICflag==1]
stat34, p34 = ttest_ind(data33, data44)
print('\nP value of t-test for drifting buoy = ',p34)
if p34 > 0.05:
 print('Probably the same distribution')
else:
 print('We reject the null hypothesis and support the alternative one so ISST and OSST probably ha
ve different distributions\n(i.e., the means are unequal). This can be an evidence that the error
between them is significant.')

data55 = mb.ISST[mb.ICflag==1]
data66 = mb.OSST[mb.ICflag==1]
stat56, p56 = ttest_ind(data55, data66)
print('\nP value of t-test for moored buoy = ',p56)
if p56 > 0.05:
 print('Probably the same distribution')
else:
 print('We reject the null hypothesis and support the alternative one so ISST and OSST probably ha
ve different distributions\n(i.e., the means are unequal). This can be an evidence that the error
between them is significant.')
```

```
P value of t-test for ship =  2.598647367503623e-33
We reject the null hypothesis and support the alternative one so ISST and OSST probably have diffe
rent distributions
(i.e., the means are unequal). This can be an evidence that the error between them is significant.

P value of t-test for drifting buoy =  1.8087169959472855e-05
We reject the null hypothesis and support the alternative one so ISST and OSST probably have diffe
rent distributions
(i.e., the means are unequal). This can be an evidence that the error between them is significant.

P value of t-test for moored buoy =  3.80785495664799e-15
We reject the null hypothesis and support the alternative one so ISST and OSST probably have diffe
rent distributions
(i.e., the means are unequal). This can be an evidence that the error between them is significant.
```

Pearson’s Test

H0: ISST and OSST are independent.

H1: ISSt and OSST are dependent.

In [106]:

```
# Example of the Pearson's Correlation test
from scipy.stats import pearsonr

datash = sh.ISST
dataash = sh.OSST
stat, psh = pearsonr(datash, dataash)
print('P value for ship = ', psh)
if psh > 0.05:
 print('Probably independent')
else:
 print('Probably dependent')

datadb = db.ISST
dataadb = db.OSST
statt, pdb = pearsonr(datadb, dataadb)
print('P value for DB = ', pdb)
if pdb > 0.05:
 print('Probably independent')
else:
 print('Probably dependent')

datamb = mb.ISST
```

```
dataamb = mb.OSST
stattt, pmb = pearsonr(datamb, dataamb)
print('P value for MB = ', pmb)
if pmb> 0.05:
 print('Probably independent')
else:
 print('Probably dependent')
```

```
P value for ship =  1.0
Probably independent
P value for DB =  0.0
Probably dependent
P value for MB =  0.0
Probably dependent
```

Analysis of Variance Test (ANOVA)

H0: the means of ISST-OSST are equal.

H1: one or more of the means of ISST-OSST are unequal.

In [35]:

```
datad = sh
datad['d'] = datad.ISST[datad.ICflag==1] - datad.OSST[datad.ICflag==1]
#datad.d
ss = datad['d'].dropna()
#ss
```

In [36]:

```
data1d = db
data1d['d'] = data1d.ISST[data1d.ICflag==1] - data1d.OSST[data1d.ICflag==1]
#data1d.d
sd = data1d['d'].dropna()
#sd
```

In [37]:

```
data2d = mb
data2d['d'] = data2d.ISST[data2d.ICflag==1] - data2d.OSST[data2d.ICflag==1]
#datad.d
sm = data2d['d'].dropna()
#sm
```

In [38]:

```
from scipy.stats import f_oneway

shd = ss

dbd = sd

mbd = sm

statd, pd = f_oneway(shd, dbd, mbd)
print('P value = ', pd)
if pd > 0.05:
 print('\033[1m' +'Probably the same distribution')
else:
 print('We reject the null hypothesis and support the alternative hypothesis since the P value is smaller than 0.05 so\nthe errors of ISST and OSST have different distributions.')
```

```
P value =  0.0
We reject the null hypothesis and support the alternative hypothesis since the P value is smaller
than 0.05 so
the errors of ISST and OSST have different distributions.
```

We can conclude that drifting buoy is the best in situ instrument to measure sea surface temperature since the error variations is less than error variations for ship and moored buoy. It is not a must since the measurements depend also on the state of the instruments so in our case we can say that. Thus, we should try to reduce the systematic errors.

In [ ]:

```
```