In [1]:

```
import pandas as pd
from pandas_profiling import ProfileReport

import altair as alt
import import_data
```

In [2]:

```
try:
    df = import_data.load_data()
except:
    import_data.download_data()
    df = import_data.load_data()

df['Date'] = pd.DatetimeIndex(df['Date'])
```

Summary of the data set

The data set is from the Canadian Ice Thickness Program. The data has been collected weekly since 1947. The program was updated in 2002, so we are only looking at data prior to the update. Ice thickness is measured to the nearest centimetre using one of two methods; special auger kit or hot wire ice thickness gauge.

Data overview

Our data set has a range of dates from 1984 - 1996. There are 195 different stations at which measurements are taken.

In [3]:

```
df
```

Out[3]:

	StationID/ID de station	Station Name/Nom de station	Date	Ice Thickness/ Épaisseur de la glace	Snow depth/Profondeur de la neige	Measurement Method/Méthode de mesure	Surface Topology/Topographie de la surface	Cracks and Leads/Fissures et chenaux
0	Q25	14A (END BECANCOUR DOCK) Q25	1984-01-07	40.0	1.0	NaN	NaN	NaN
1	Q25	14A (END BECANCOUR DOCK) Q25	1984-01-16	49.0	20.0	NaN	NaN	NaN

	StationID/ID de station	Station Name/Nom de station	Date	Ice Thickness/ Épaisseur de la glace	Snow depth/Profondeur de la neige	Measurement Method/Méthode de mesure	Surface Topology/Topographie de la surface	Cracks and Leads/Fissures et chenaux
2	Q25	14A (END BECANCOUR DOCK) Q25	1984-01-21	42.0	8.0	NaN	NaN	NaN
3	Q25	14A (END BECANCOUR DOCK) Q25	1984-01-28	43.0	20.0	NaN	NaN	NaN
4	Q25	14A (END BECANCOUR DOCK) Q25	1984-02-04	41.0	22.0	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
51186	YZF	YELLOWKNIFE YZF	1996-03-29	140.0	18.0	1.0	0.0	0.0
51187	YZF	YELLOWKNIFE YZF	1996-04-05	136.0	24.0	1.0	0.0	0.0
51188	YZF	YELLOWKNIFE YZF	1996-04-12	144.0	14.0	1.0	0.0	0.0
51189	YZF	YELLOWKNIFE YZF	1996-04-19	143.0	10.0	1.0	0.0	0.0
51190	YZF	YELLOWKNIFE YZF	1996-04-26	154.0	4.0	1.0	0.0	0.0

51191 rows × 8 columns

```
In [4]: df["Station Name/Nom de station"].value_counts()
```

```
Out[4]: EUREKA WEU                       1731
        RESOLUTE YRB                     1641
        ALERT YLT                        1434
        CAMBRIDGE BAY YCB                1389
        MOULD BAY YMD                    1388
                                         ...
        ST. PETERS BAY YG5                  4
```

```
P23 (NORTHWEST SECTION) Q23       2
P24 (OFF PUBLIC DOCK) Q24         2
SUMMERSIDE YG1                    2
SOURIS YG6                        2
Name: Station Name/Nom de station, Length: 195, dtype: int64
```

Data value ranges

We have 5112 ice thickness measurements. The mean ice thickness over all dates is ~93.26. The standard deviation is ~57.63, and the measurements range from 0 - 345.

In [5]:

```
df.describe()
```

Out[5]:

	Ice Thickness/ Épaisseur de la glace	Snow depth/Profondeur de la neige	Measurement Method/Méthode de mesure	Surface Topology/Topographie de la surface	Cracks and Leads/Fissures et chenaux
count	51125.000000	48652.000000	15604.000000	15425.000000	15428.000000
mean	93.257643	14.493978	0.981287	0.599481	0.436349
std	57.632578	13.532427	0.144664	1.582073	0.669096
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	46.000000	4.000000	1.000000	0.000000	0.000000
50%	79.000000	10.000000	1.000000	0.000000	0.000000
75%	135.000000	21.000000	1.000000	0.000000	1.000000
max	345.000000	152.000000	3.000000	9.000000	9.000000

Data types and completeness

Each row has a `Date`, `Station ID`, and a `Station Name`. There are 66 rows that are missing an `Ice Thickness` measurement.

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51191 entries, 0 to 51190
Data columns (total 8 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   StationID/ID de station                     51191 non-null  object
 1   Station Name/Nom de station                  51191 non-null  object
 2   Date                                         51191 non-null  datetime64[ns]
 3   Ice Thickness/Épaisseur de la glace          51125 non-null  float64
 4   Snow depth/Profondeur de la neige            48652 non-null  float64
 5   Measurement Method/Méthode de mesure         15604 non-null  float64
 6   Surface Topology/Topographie de la surface  15425 non-null  float64
 7   Cracks and Leads/Fissures et chenaux         15428 non-null  float64
dtypes: datetime64[ns](1), float64(5), object(2)
memory usage: 3.1+ MB
```

Variables and interactions

Most of the rows have the same `Measurement Method`, but there are some that are missing the method or have a different method. We will need to make sure we are only using rows with the same measurement method in our sample.

In [7]:

```
df["Measurement Method/Méthode de mesure"].value_counts()
```

Out[7]:

```
1.0    15278
0.0      310
2.0       14
3.0        2
Name: Measurement Method/Méthode de mesure, dtype: int64
```

In [8]:

```
ProfileReport(df)
```

Overview

Dataset statistics

Number of variables	8
Number of observations	51191

Variable types

NUM	4
CAT	3

Missing cells	109721
Missing cells (%)	26.8%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	3.1 MiB
Average record size in memory	64.0 B

DATE	1

Warnings

`StationID/ID de station` has a high cardinality: 195 distinct values	**High cardinality**
`Station Name/Nom de station` has a high cardinality: 195 distinct values	**High cardinality**
`Snow depth/Profondeur de la neige` has 2539 (5.0%) missing values	**Missing**
`Measurement Method/Méthode de mesure` has 35587 (69.5%) missing values	**Missing**
`Surface Topology/Topographie de la surface` has 35766 (69.9%) missing values	**Missing**
`Cracks and Leads/Fissures et chenaux` has 35763 (69.9%) missing values	**Missing**
`Snow depth/Profondeur de la neige` has 6178 (12.1%) zeros	**Zeros**
`Surface Topology/Topographie de la surface` has 12955 (25.3%) zeros	**Zeros**
`Cracks and Leads/Fissures et chenaux` has 9808 (19.2%) zeros	**Zeros**

```
Out[8]:
```

Exploratory analysis of Ice Thickness

To better understand our data and to determine how to sample it we explored:

- Number of ice thickness measurements per date
- Number of stations per date

- General change in ice thickness over time
- Distribution of ice thickness over all time
- Distribution of ice thickness for each date of interest
- Determine if there are outliers in the ice thickness measurements

We removed records with `Measurement Method` not equal to 1 in order to make sure the measurement method we are looking at is consistent. We also removed all records missing an `Ice Thickness` measurement.

In [9]:

```
df_filtered = df.copy()
df_filtered = df_filtered.rename(columns={
    "StationID/ID de station" : "station_id",
    "Station Name/Nom de station" : "station_name",
    "Date" : "date",
    "Ice Thickness/Épaisseur de la glace" : "ice_thickness",
    "Snow depth/Profondeur de la neige" : "snow_depth",
    "Measurement Method/Méthode de mesure" : "measurement_method",
    "Surface Topology/Topographie de la surface" : "surface_topology",
    "Cracks and Leads/Fissures et chenaux" : "cracks_leads"
})

df_filtered = df_filtered[df_filtered["measurement_method"] == 1]
df_filtered = df_filtered[df_filtered["ice_thickness"] > 0]
df_filtered.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 15253 entries, 233 to 51190
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   station_id          15253 non-null  object
 1   station_name        15253 non-null  object
 2   date                15253 non-null  datetime64[ns]
 3   ice_thickness       15253 non-null  float64
 4   snow_depth          15056 non-null  float64
 5   measurement_method  15253 non-null  float64
 6   surface_topology    15101 non-null  float64
 7   cracks_leads        15104 non-null  float64
dtypes: datetime64[ns](1), float64(5), object(2)
memory usage: 1.0+ MB
```

Number of ice thickness measurements

We looked at number of ice thickness measurements per day, month, and year. Each year, January - March had the largest number of measurements. July - September had the smallest number of measurements, with no measurements taken in August each year.

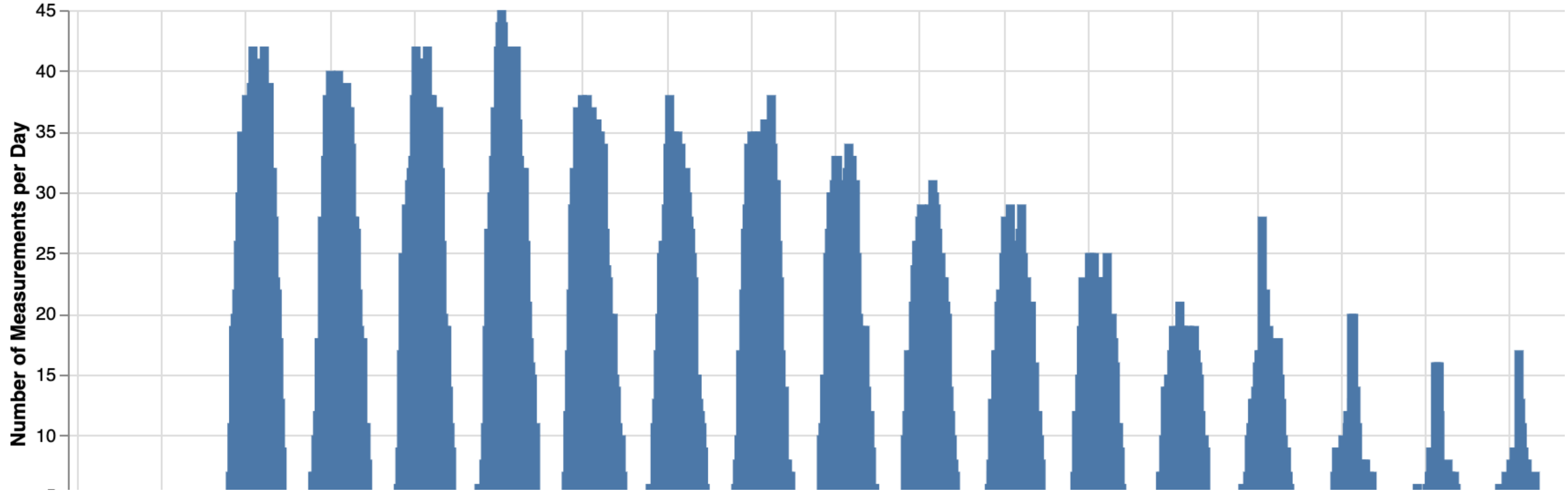
Presumably this is because the ice melts each summer.

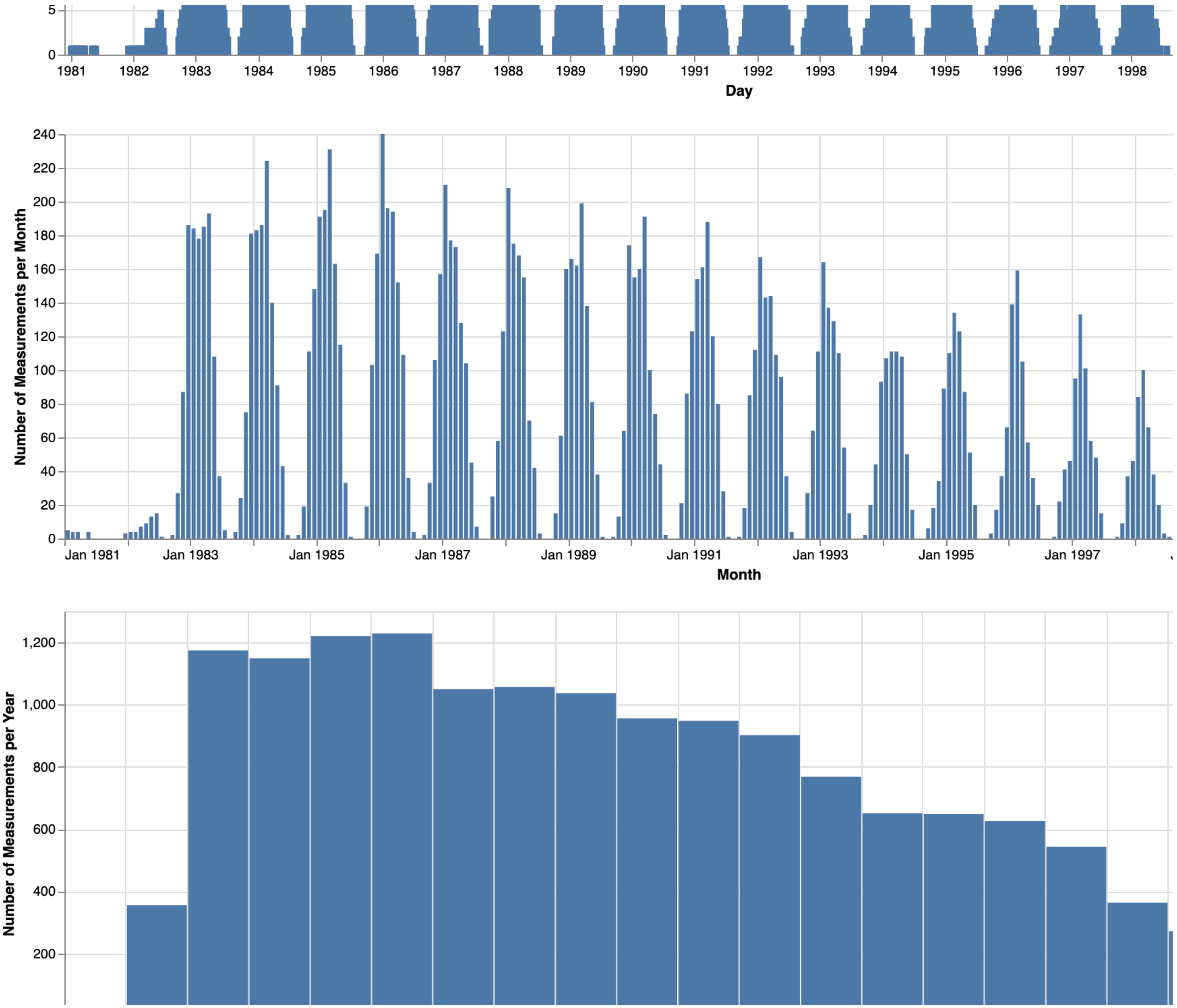
In [10]:

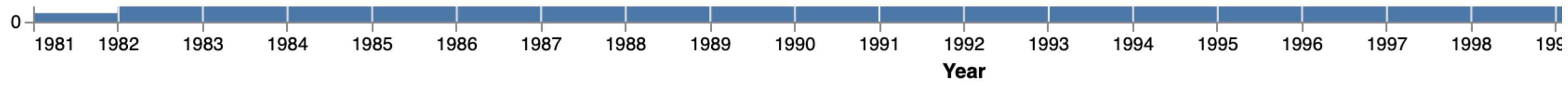
```
alt.data_transformers.disable_max_rows()

count_date_chart = alt.Chart(df_filtered).mark_bar().encode(
    x = alt.X("date", title="Day"),
    y = alt.Y("count()", title="Number of Measurements per Day"),
    tooltip = ["date", "count()"]
).properties(
    width=1000
)

count_month_chart = count_date_chart.encode(
    x = alt.X("yearmonth(date)", title="Month"),
    y = alt.Y("count()", title="Number of Measurements per Month"),
    tooltip = ["yearmonth(date)", "count()"]
)

count_year_chart = count_date_chart.encode(
    x = alt.X("year(date)", title="Year"),
    y = alt.Y("count()", title="Number of Measurements per Year"),
    tooltip = ["year(date)", "count()"]
)

(count_date_chart & count_month_chart & count_year_chart).properties(
    title="Number of Ice Thickness Measurements by Date",
)
```

Out[10]:

Number of Ice Thickness Measurements by Date




5
0
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
Day
Number of Measurements per Month
240
220
200
180
160
140
120
100
80
60
40
20
0
Jan 1981
Jan 1983
Jan 1985
Jan 1987
Jan 1989
Jan 1991
Jan 1993
Jan 1995
Jan 1997
Month
Number of Measurements per Year
1,200
1,000
800
600
400
200



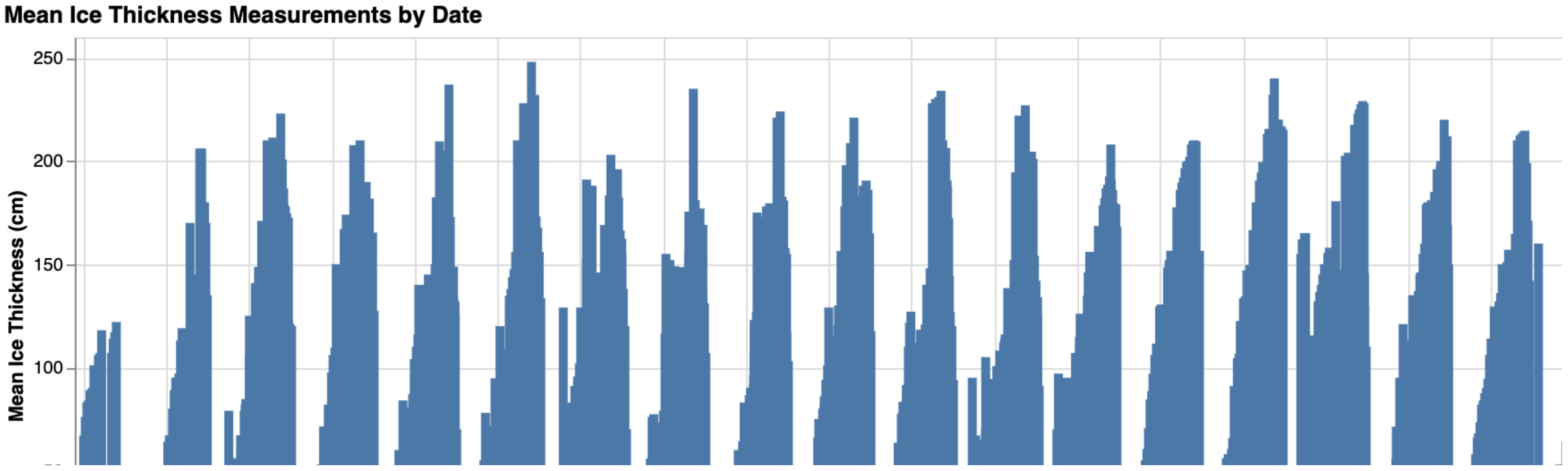
Mean ice thickness measurements by date

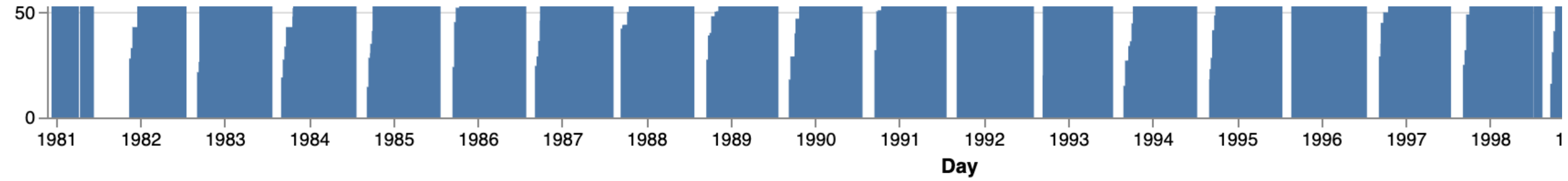
We looked at mean ice thickness measurements per day, month, and year. Each year, May - June had the highest mean ice thickness measurements. September had the smallest number of measurements, with no measurements taken in August each year. The mean ice thickness meausurements fluctuate year over year, but they seem to typically be around 100 cm.

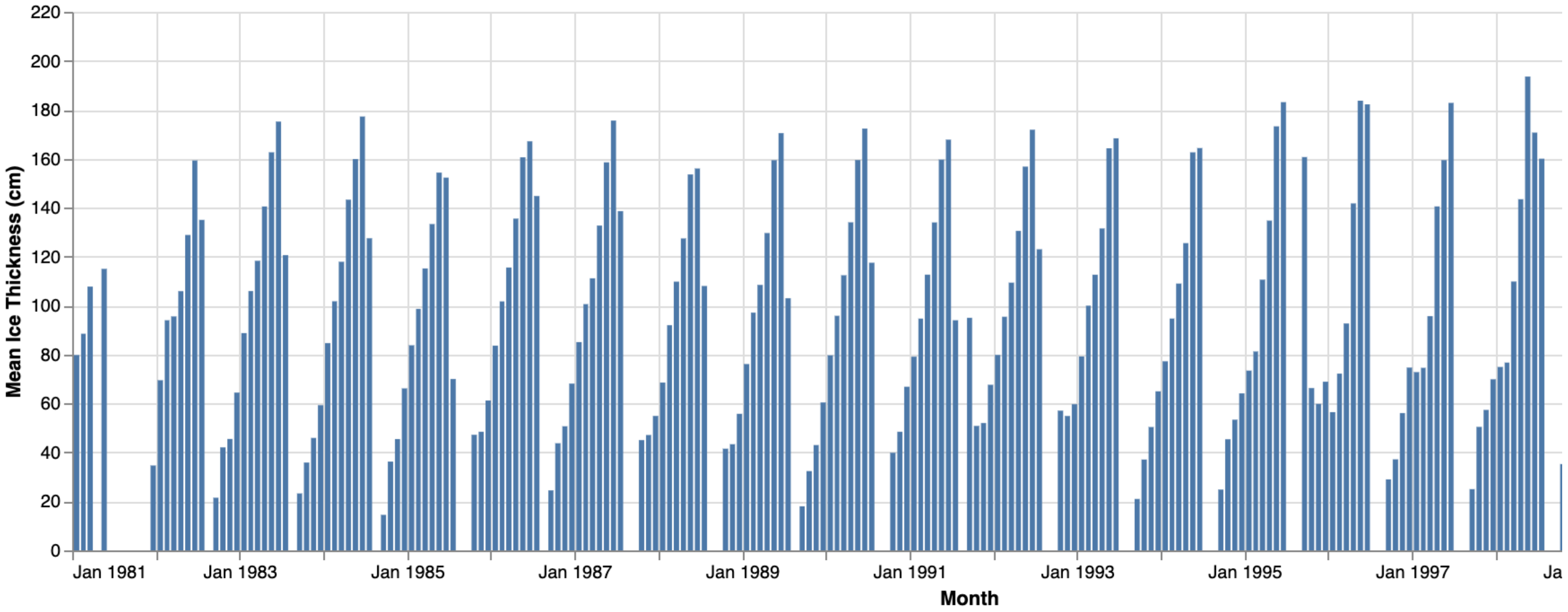
In [11]:

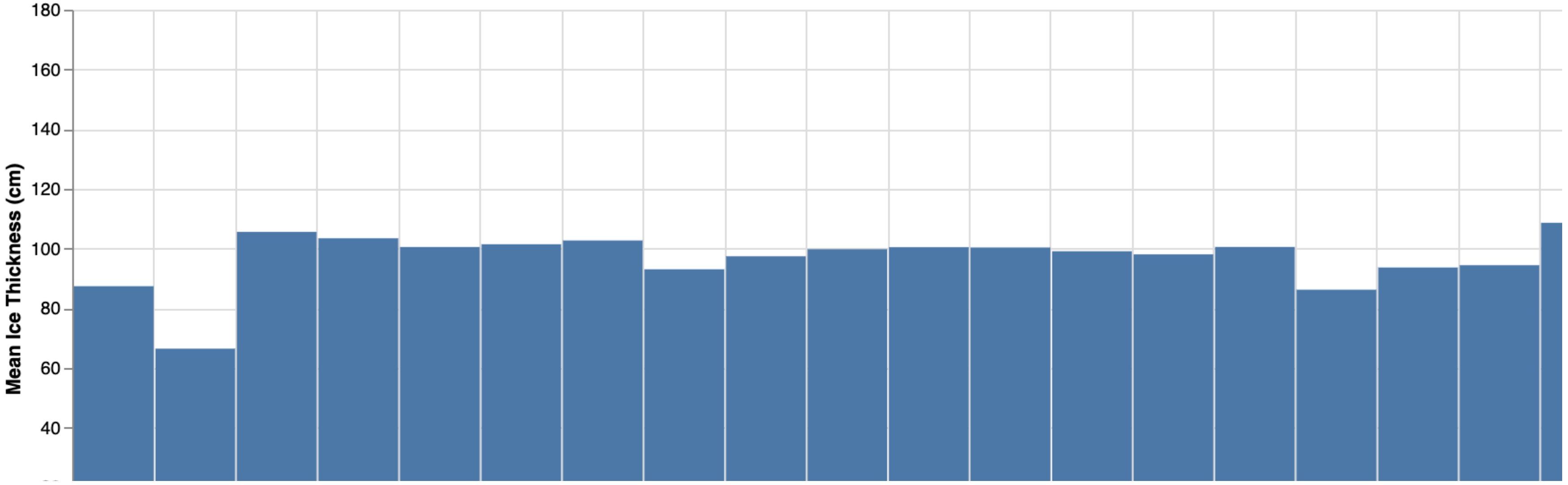
```
measurement_date_chart = count_date_chart.encode(
    y = alt.Y("mean(ice_thickness)", title="Mean Ice Thickness (cm)"),
    tooltip = ["date", "mean(ice_thickness)"]
)

measurement_month_chart = count_month_chart.encode(
    y = alt.Y("mean(ice_thickness)", title="Mean Ice Thickness (cm)"),
    tooltip = ["yearmonth(date)", "mean(ice_thickness)"]
)

measurement_year_chart = count_year_chart.encode(
    y = alt.Y("mean(ice_thickness)", title="Mean Ice Thickness (cm)"),
    tooltip = ["year(date)", "mean(ice_thickness)"]
)

(measurement_date_chart & measurement_month_chart & measurement_year_chart).properties(
    title="Mean Ice Thickness Measurements by Date",
)
```

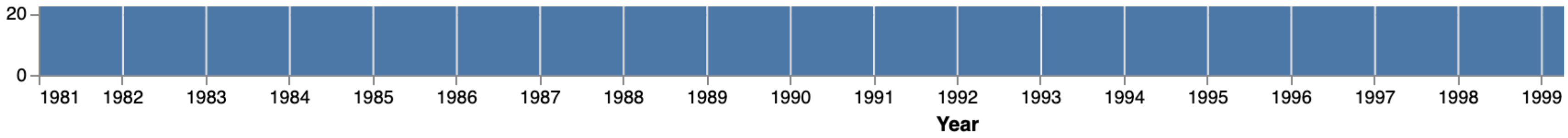
Out[11]:




50
0
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1
Day


220
200
180
160
140
120
100
80
60
40
20
0
Mean Ice Thickness (cm)
Jan 1981
Jan 1983
Jan 1985
Jan 1987
Jan 1989
Jan 1991
Jan 1993
Jan 1995
Jan 1997
Ja
Month

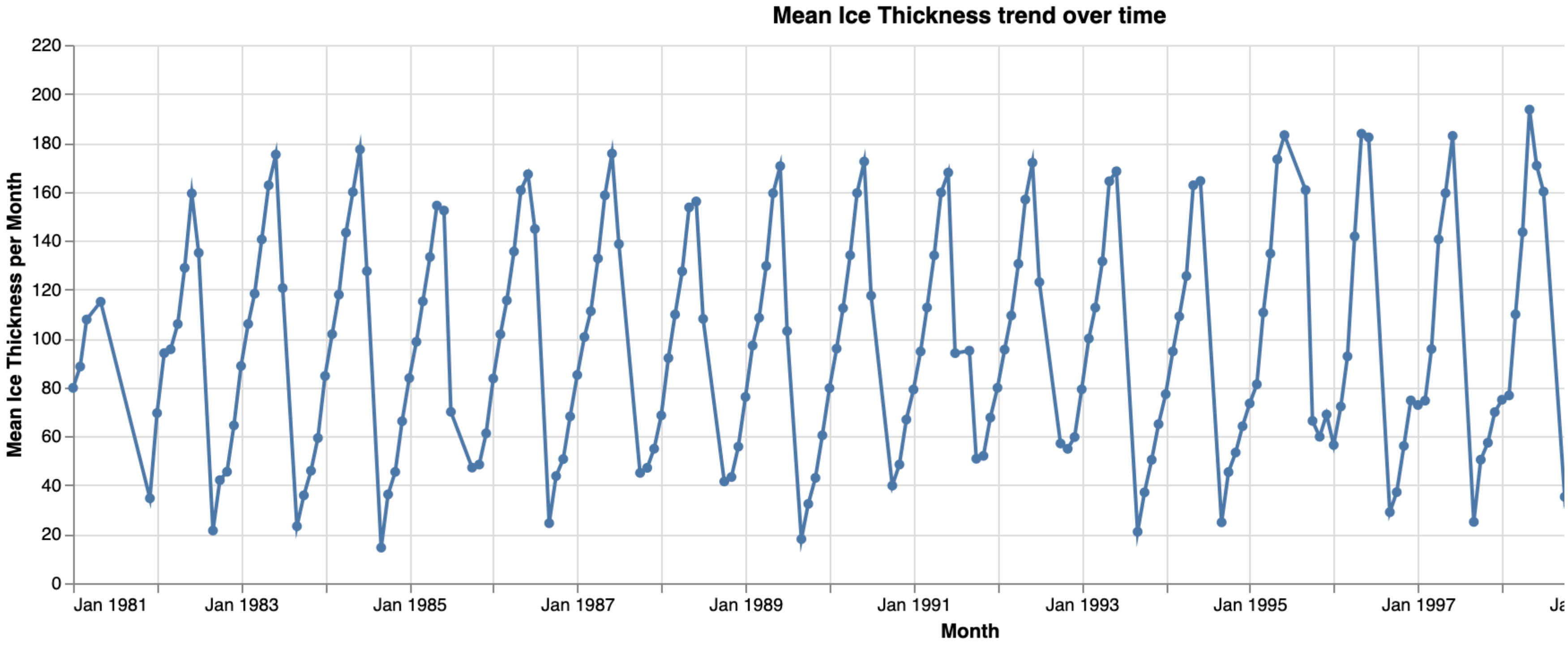

180
160
140
120
100
80
60
40
Mean Ice Thickness (cm)



In [12]:

```
ice_chart = alt.Chart(df_filtered, title="Mean Ice Thickness trend over time").mark_line().encode(
    x = alt.X("yearmonth(date)", title="Month"),
    y = alt.Y("mean(ice_thickness)", title="Mean Ice Thickness per Month"),
    tooltip = ["yearmonth(date)", "mean(ice_thickness)"]
).properties(
    width=1000
)

ice_chart + ice_chart.mark_circle()
```

Out[12]:



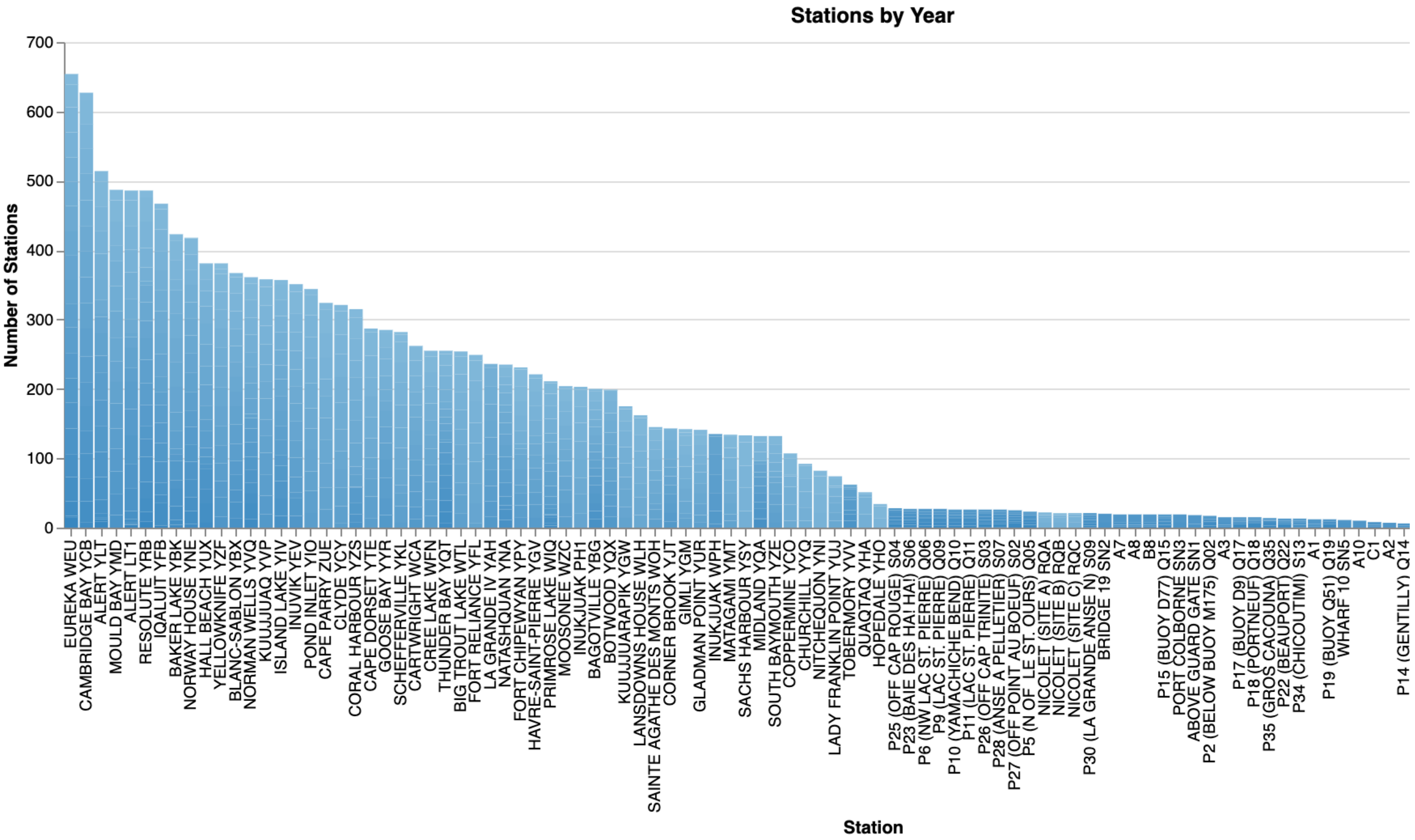
Number of Stations

Stations vary over the years but seem relatively consistent. Some stations seem to be replaced over time, but the stations with the majority of measurements have records for each year.

In [13]:

```
# Number of stations per year
alt.Chart(df_filtered, title = "Stations by Year").mark_bar().encode(
    x = alt.X("station_name", title="Station", sort = '-y'),
    y = alt.Y("count()", title="Number of Stations"),
    color = "year(date)",
    tooltip = ["year(date)", "count()"]
).properties(
    width = 1000
)
```

Out[13]:



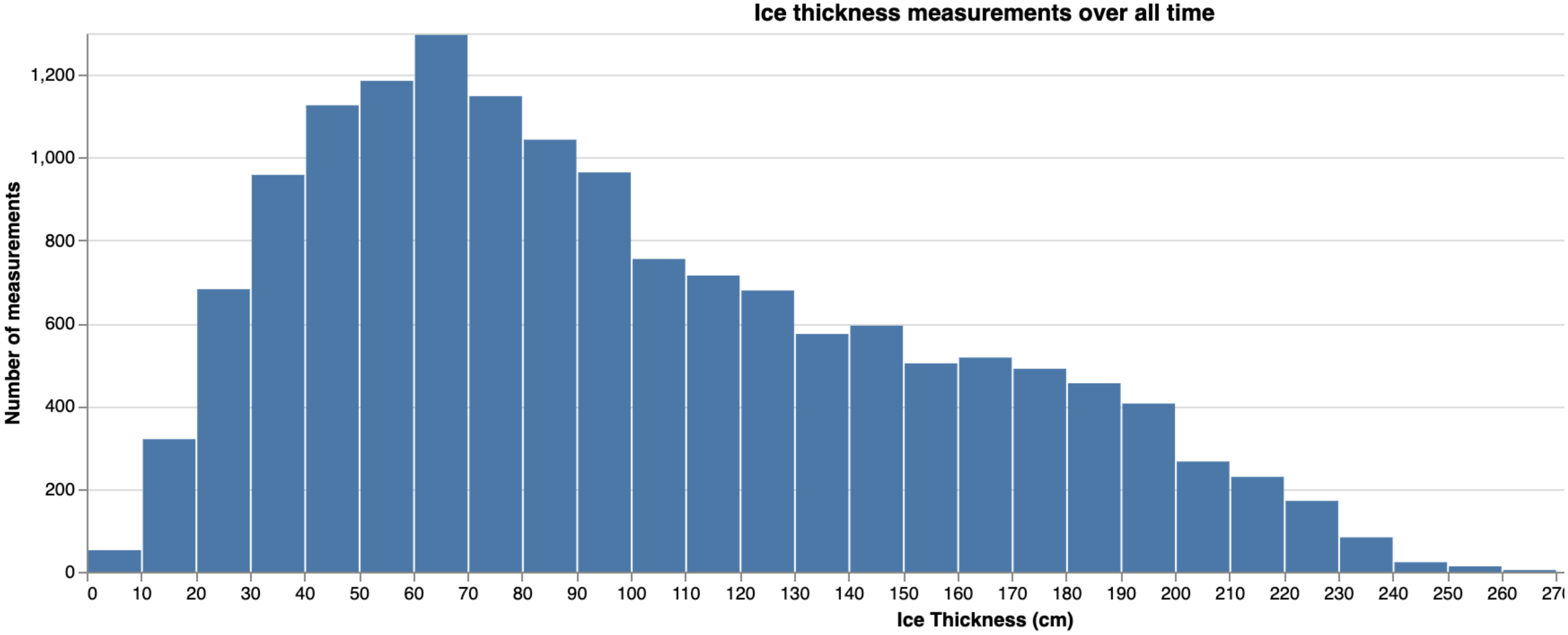
Ice thickness distribution

We looked at the distribution of thickness measurements over all time, by month, and by year. The distribution over all time and the distributions per year are right skewed. The shape of the distributions by month vary from month to month.

In [14]:

```
# Distribution of ice thickness over all time

ice_histogram = alt.Chart(df_filtered, title="Ice thickness measurements over all time").mark_bar().encode(
    x = alt.X("ice_thickness", title="Ice Thickness (cm)", bin=alt.Bin(maxbins=40)),
    y = alt.Y("count()", title="Number of measurements"),

).properties(
    width=1000
)

ice_histogram
```

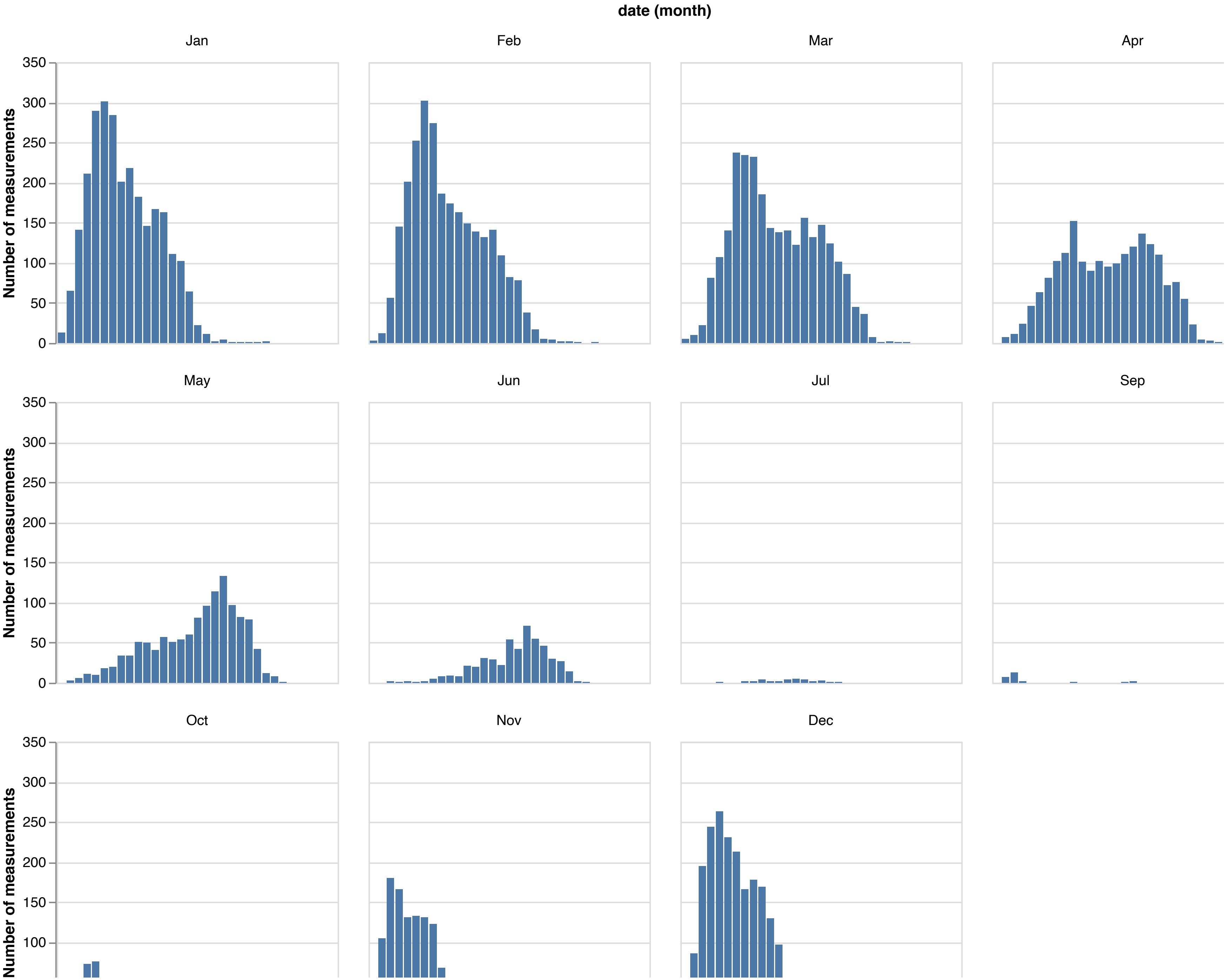
Out[14]:

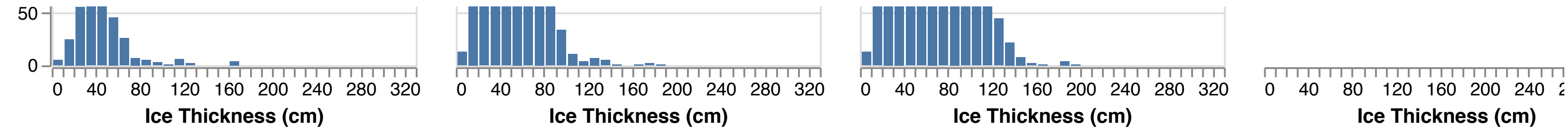


In [15]:

```
ice_histogram.properties(
    width=200,
    height=200
).facet(
    "month(date)",
    columns = 4
)
```

Out[15]:

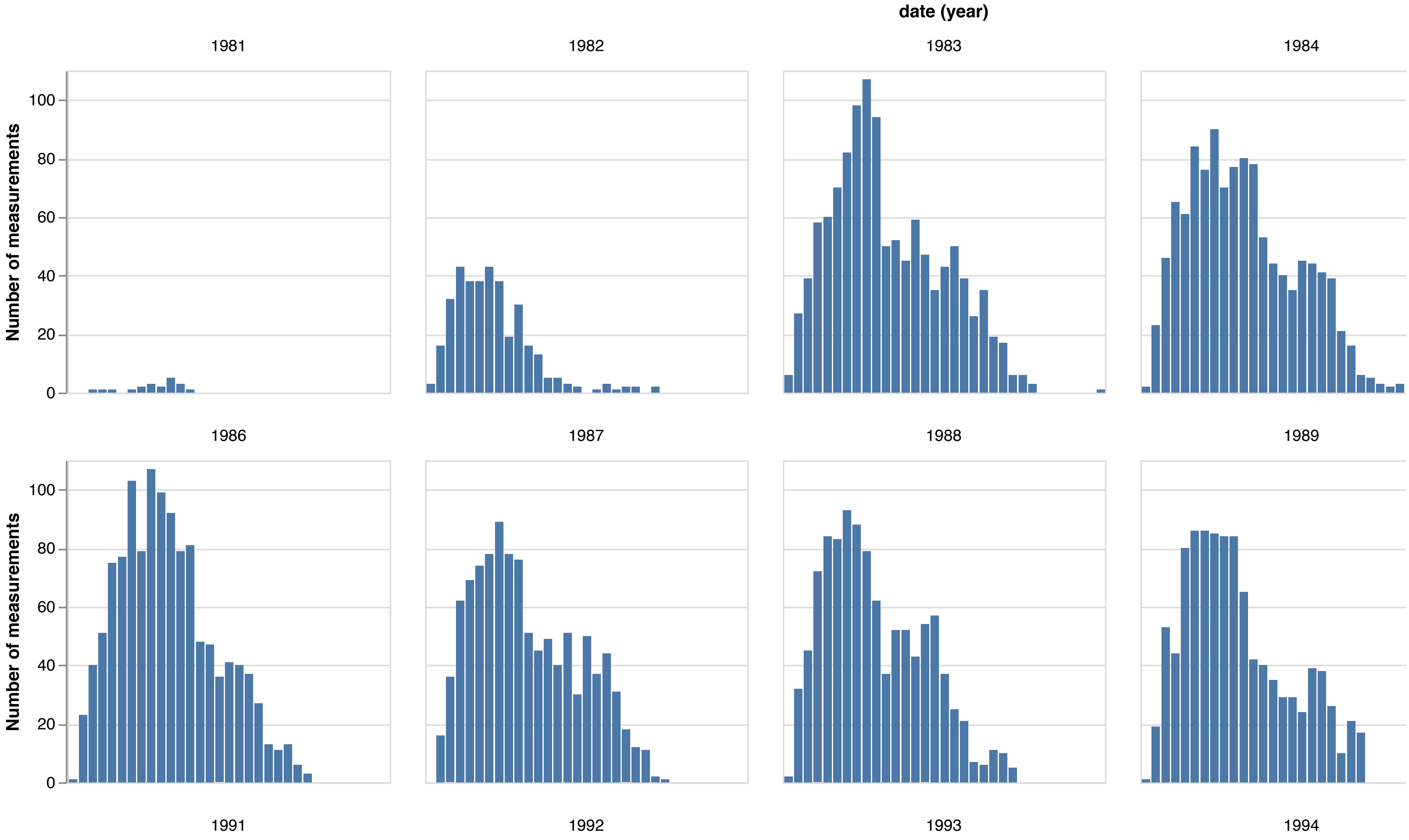


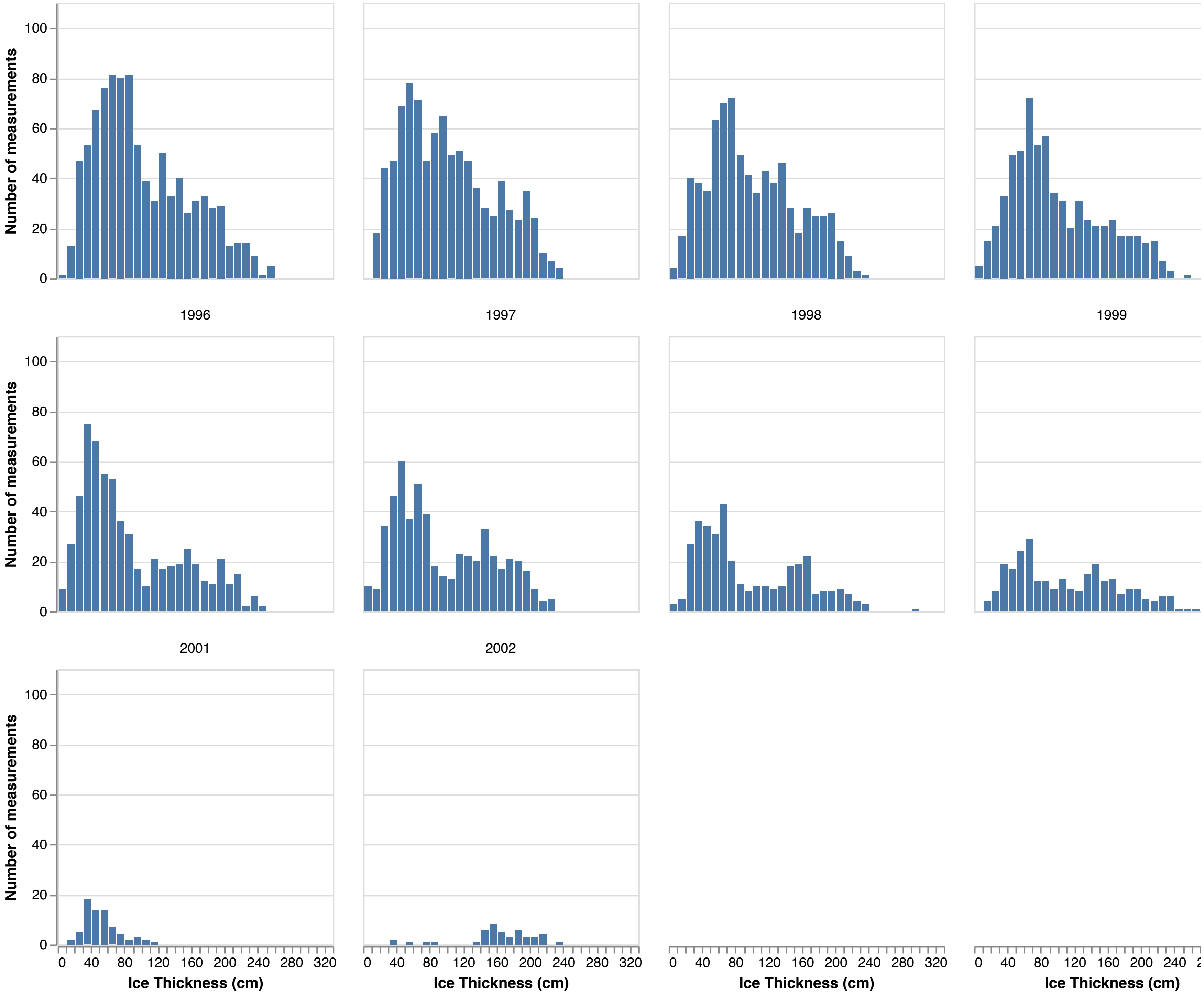


In [16]:

```
ice_histogram.properties(
    width=200,
    height=200
).facet(
    "year(date)",
    columns = 5
)
```

Out[16]:




Number of measurements
100
80
60
40
20
0
1996
1997
1998
1999
2001
2002
0
40
80
120
160
200
240
280
320
Ice Thickness (cm)

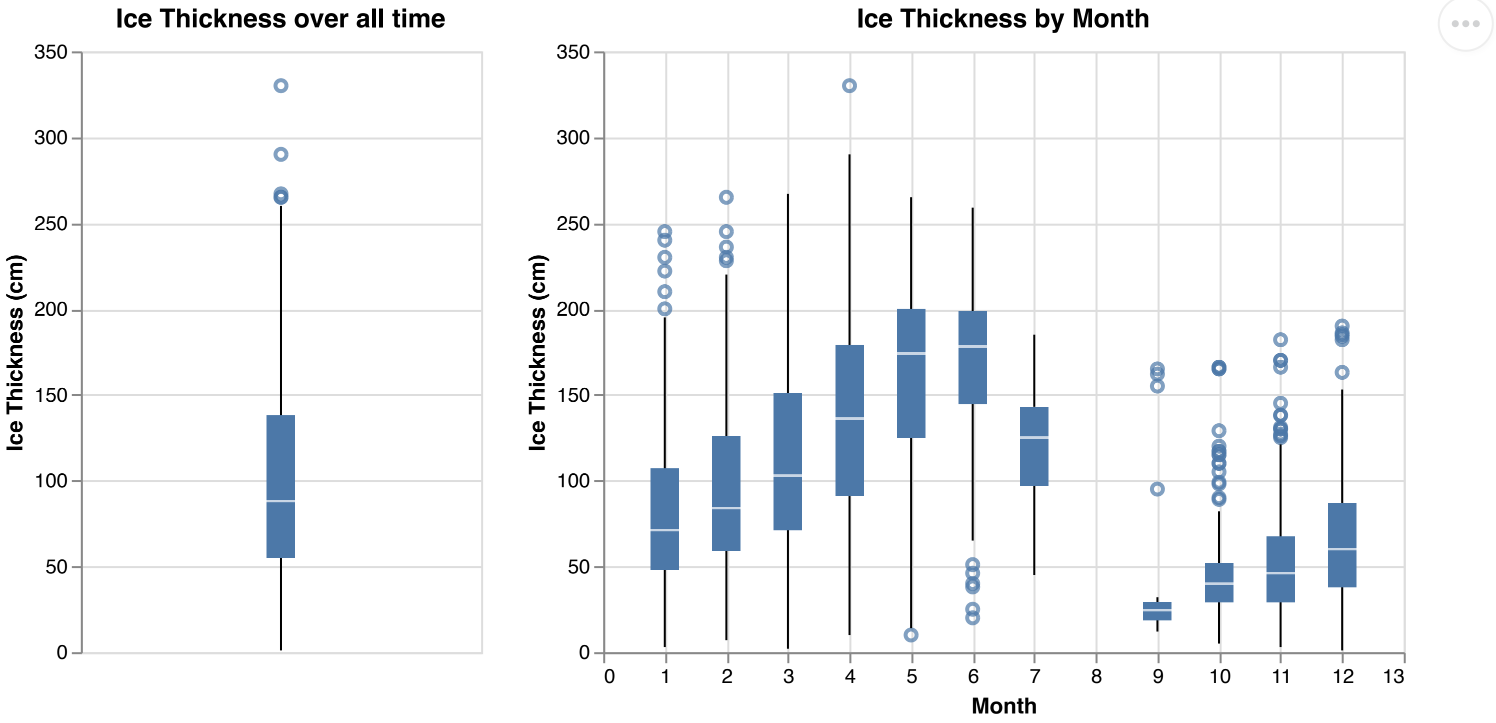
Looking for outliers

We looked at boxplots over all time and by month, as well as over all time per location. There are not many observations that seem out of place. The distributions vary by month, as we saw earlier as well.

In [17]:

```
df_filtered["month"] = df_filtered["date"].dt.month

month_boxplot = alt.Chart(df_filtered, title="Ice Thickness by Month").mark_boxplot().encode(
    y=alt.Y("ice_thickness", title="Ice Thickness (cm)"),
    x=alt.X("month", title="Month"),
    tooltip=["ice_thickness", "date", "station_name"]
)

all_time_boxplot = alt.Chart(df_filtered, title="Ice Thickness over all time").mark_boxplot().encode(
    y=alt.Y("ice_thickness", title="Ice Thickness (cm)"),
    tooltip=["ice_thickness", "date", "station_name"]
).properties(
    width=200
)

all_time_boxplot | month_boxplot
```

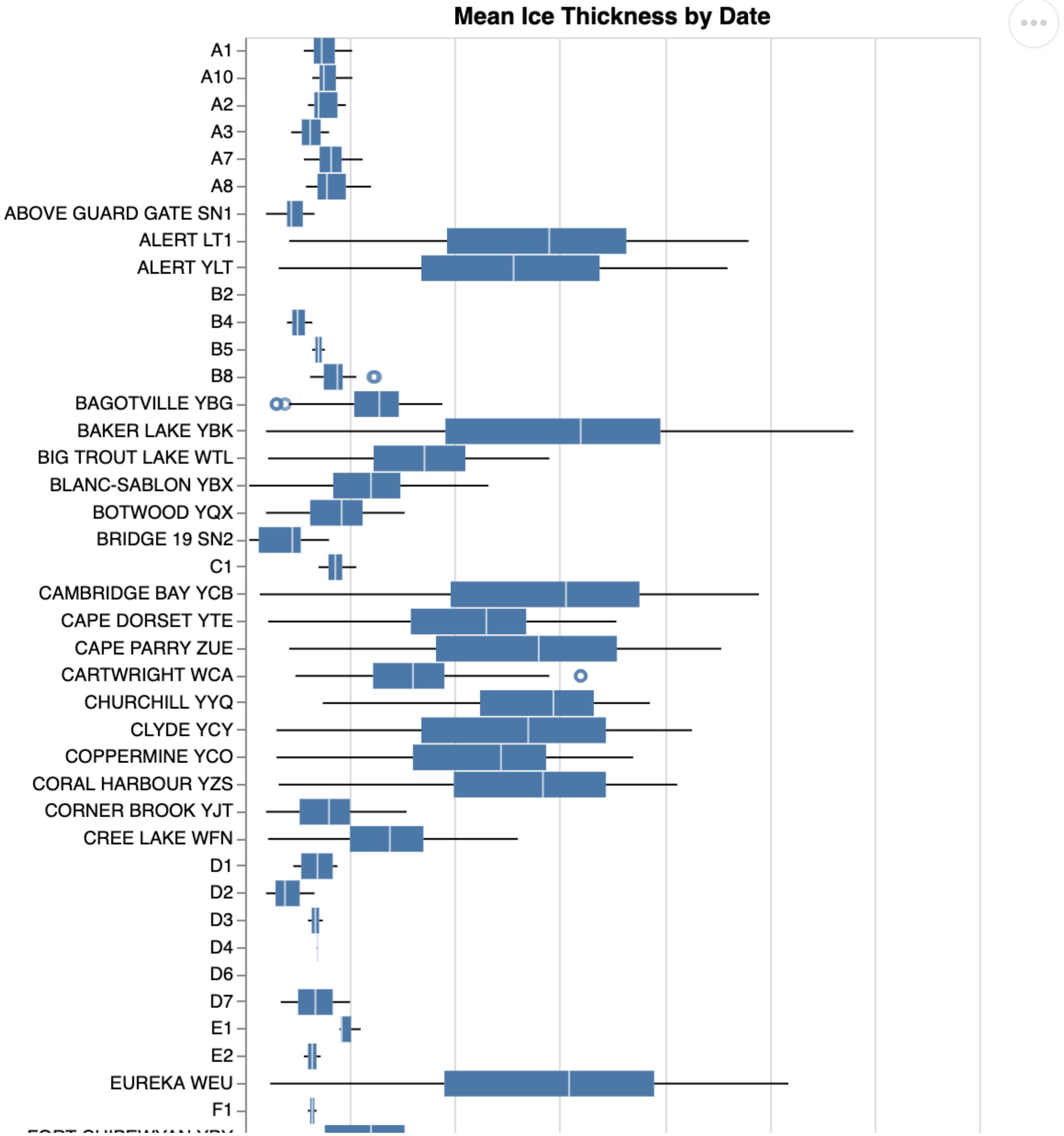
Out[17]:

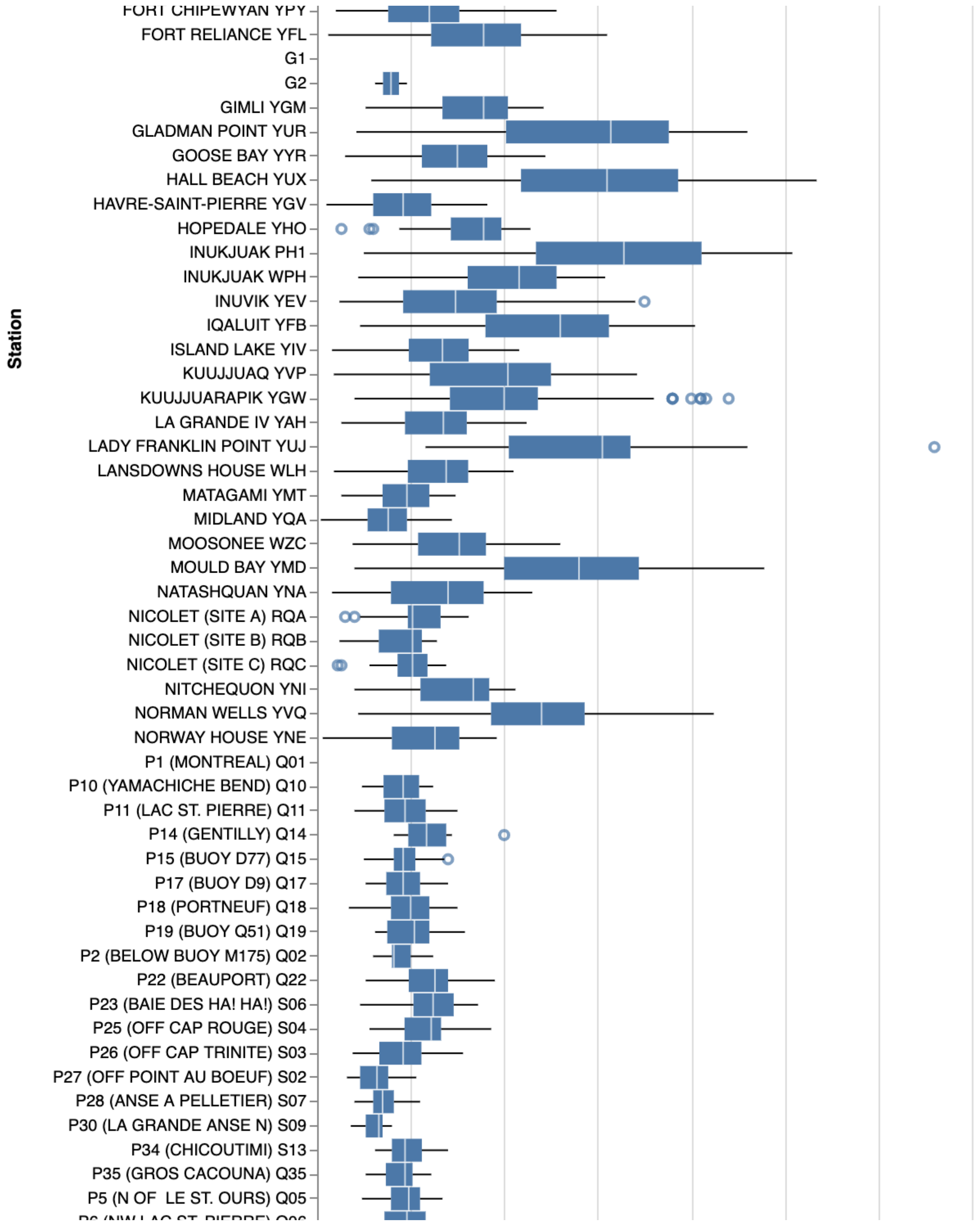


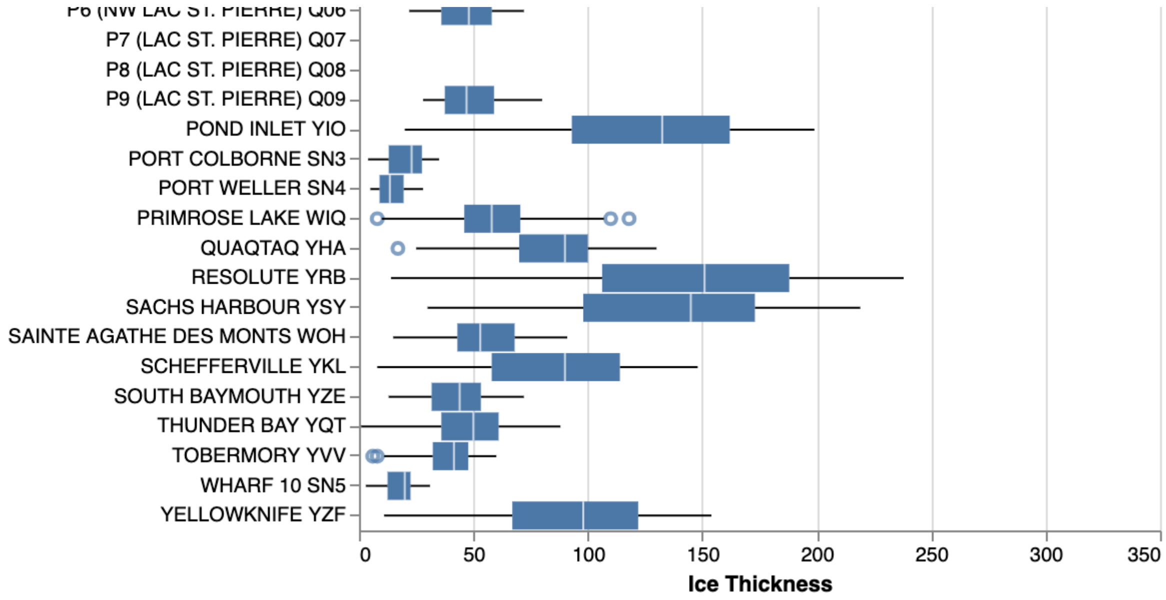
In [18]:

```
alt.Chart(df_filtered, title="Mean Ice Thickness by Date").mark_boxplot().encode(
    x = alt.X("ice_thickness", title="Ice Thickness"),
    y = alt.Y("station_name", title="Station"),
    tooltip=["ice_thickness", "date"]
).properties(
    height=1600
)
```

Out[18]:




Station
FORT CHIPEWYAN YPY
FORT RELIANCE YFL
G1
G2
GIMLI YGM
GLADMAN POINT YUR
GOOSE BAY YYR
HALL BEACH YUX
HAVRE-SAINT-PIERRE YGV
HOPEDALE YHO
INUKJUAK PH1
INUKJUAK WPH
INUVIK YEV
IQALUIT YFB
ISLAND LAKE YIV
KUUJJUAQ YVP
KUUJJUARAPIK YGW
LA GRANDE IV YAH
LADY FRANKLIN POINT YUJ
LANSDOWNS HOUSE WLH
MATAGAMI YMT
MIDLAND YQA
MOOSONEE WZC
MOULD BAY YMD
NATASHQUAN YNA
NICOLET (SITE A) RQA
NICOLET (SITE B) RQB
NICOLET (SITE C) RQC
NITCHEQUON YNI
NORMAN WELLS YVQ
NORWAY HOUSE YNE
P1 (MONTREAL) Q01
P10 (YAMACHICHE BEND) Q10
P11 (LAC ST. PIERRE) Q11
P14 (GENTILLY) Q14
P15 (BUOY D77) Q15
P17 (BUOY D9) Q17
P18 (PORTNEUF) Q18
P19 (BUOY Q51) Q19
P2 (BELOW BUOY M175) Q02
P22 (BEAUPORT) Q22
P23 (BAIE DES HA! HA!) S06
P25 (OFF CAP ROUGE) S04
P26 (OFF CAP TRINITE) S03
P27 (OFF POINT AU BOEUF) S02
P28 (ANSE A PELLETIER) S07
P30 (LA GRANDE ANSE N) S09
P34 (CHICOUTIMI) S13
P35 (GROS CACOUNA) Q35
P5 (N OF LE ST. OURS) Q05
P6 (NW LAC ST. PIERRE) Q06


P6 (NW LAC ST. PIERRE) Q06
P7 (LAC ST. PIERRE) Q07
P8 (LAC ST. PIERRE) Q08
P9 (LAC ST. PIERRE) Q09
POND INLET YIO
PORT COLBORNE SN3
PORT WELLER SN4
PRIMROSE LAKE WIQ
QUAQTAQ YHA
RESOLUTE YRB
SACHS HARBOUR YSY
SAINTE AGATHE DES MONTS WOH
SCHEFFERVILLE YKL
SOUTH BAYMOUTH YZE
THUNDER BAY YQT
TOBERMORY YVV
WHARF 10 SN5
YELLOWKNIFE YZF
0
50
100
150
200
250
300
350
Ice Thickness