Exploratory Data Analysis

```
library(data.table)
library(lubridate)
library(sf)
library(sp)
library(lubridate)
library(gridExtra)
library(MASS)
library(mgcv)
library(Metrics)
library(tictoc)
library(stats)
library(tidyverse)
library(ggplot2)
library(forecast)
library(tictoc)
library(stringr)
library(mapdata)
library(mapview)
library(leaflet)
library(dplyr)
library(mde)
```

We essentially have 6 data sources that we want to explore / validate. The first data source is the tide data which Carling has compiled via TideGauge_DailyData.rds. To start, we focused only on the Atlantic City location since this is the source that is referenced in the Meyer paper. In the future, we may consider replacing the Atlantic City data with Cape May data since it is closer to the ocean. As far as I know, there are no plots or references in the Meyer paper that we can use to validate whether the tide data we are working with is reasonably similar to the data they were using.

1. Tide Data

```
tide_data <- readRDS('../data/raw/TideGauge_DailyData.rds')
tide_data = as.data.table(tide_data)
```

```
## Warning in as.data.table.list(tide_data): Item 1 has 384286 rows but longest
## item has 562659; recycled with remainder.
```

```
## Warning in as.data.table.list(tide_data): Item 2 has 405165 rows but longest
## item has 562659; recycled with remainder.
```

```
## Warning in as.data.table.list(tide_data): Item 3 has 545020 rows but longest
## item has 562659; recycled with remainder.
```

```
## Warning in as.data.table.list(tide_data): Item 4 has 306335 rows but longest
## item has 562659; recycled with remainder.
```

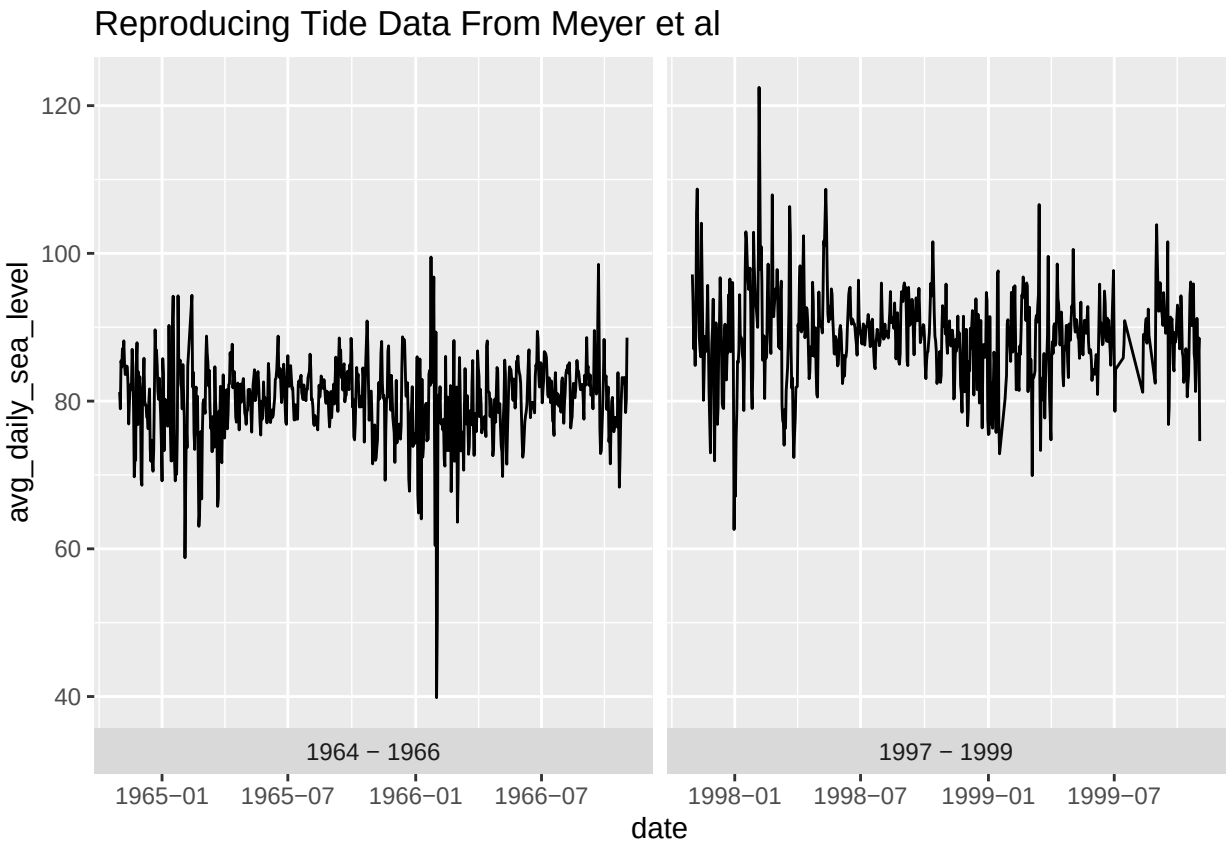
```
#Time Range 1: November 1, 1964 to November 1, 1966
#Time Range 2: November 1997 to November 1999)
```

```
relevant_tide_dat = tide_data[(AtlanticCity.datetimestamp <= '1999-11-01' & AtlanticCity.datetimestamp >
                                (AtlanticCity.datetimestamp >= '1964-11-01' & AtlanticCity.datetimestamp

# cape may sealevel
# fort pulasaki sealevel
# springmaid pier sealevel
# charleston
relevant_tide_dat$AtlanticCity.date = as.Date(relevant_tide_dat$AtlanticCity.date)
meyer_tide_dat = relevant_tide_dat[, c('AtlanticCity.date', 'AtlanticCity.sea_level')] %>%
  group_by(AtlanticCity.date) %>%
  summarise(avg_daily_sea_level = mean(AtlanticCity.sea_level))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
setnames(meyer_tide_dat, 'AtlanticCity.date', 'date')
meyer_tide_dat = as.data.table(meyer_tide_dat)[, time_range := ifelse(date > '1967-01-01', '1997 - 1999

# TIDE TS Plot
ggplot(meyer_tide_dat, aes(x = date, y = avg_daily_sea_level))+
  facet_wrap(~time_range, scales = "free_x", switch = 'x')+
  geom_line()+
  ggtitle('Reproducing Tide Data From Meyer et al')
```

```
## Warning: 'switch' is deprecated.
## Use 'strip.position' instead.
## See help("Deprecated")
```



DISCHARGE (INFLOW) DATA

We have two sources of discharge (aka streamflow aka inflow) data: Delaware River and Schulkill River. The Delware river data contains discharge data in cubic feet per second. Figure 2 in the Meyer paper plots average inflows per month at both locations. From an initial look at our data compared to the Meyer data, it appears that the seasonal patterns are non-trivially different. Our data suggests peak inflows in March while figure 2 suggests peak inflows in June. It is unclear exactly which time period the Meyer authors used to contruct Figure 2.

Next Steps: Hopefully we can find a way to align our values with what we see in Figure 2, but I don't have any ideas on how to do this other than trying to plot different time frames with trial and error?

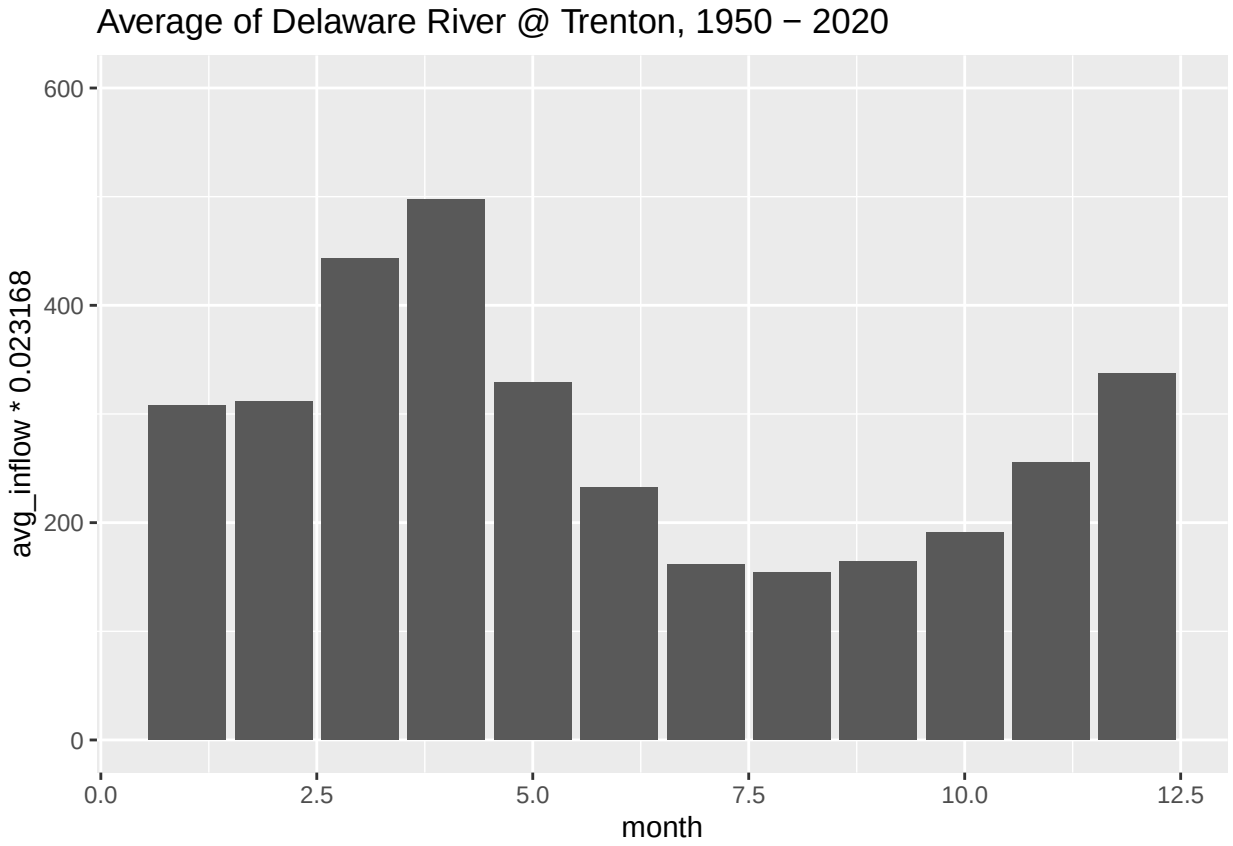
2. Delaware River at Trenton

Load in the text file as it appears in box. Calculate the % of missing values per each column to get a sense of where the useful data is. In most of these txt files there are 40+ columns with only 5 or 6 columns worth of filled in data values. Once we figure out the index of the columns with data, we open the txt file and match the codes at those indices with the descriptions to figure out what each filled-in column represents. It appears that only columns 1, 2, 3, 28, and 29 contain non-null data. According to the codes in this text file, those column names correspond to 'agency_cd', 'site_no', 'date', 'Discharge, cubic feet per second (Mean)', and a "cd" value...). Next we rename the columns so that we can extract the subset of useful data.

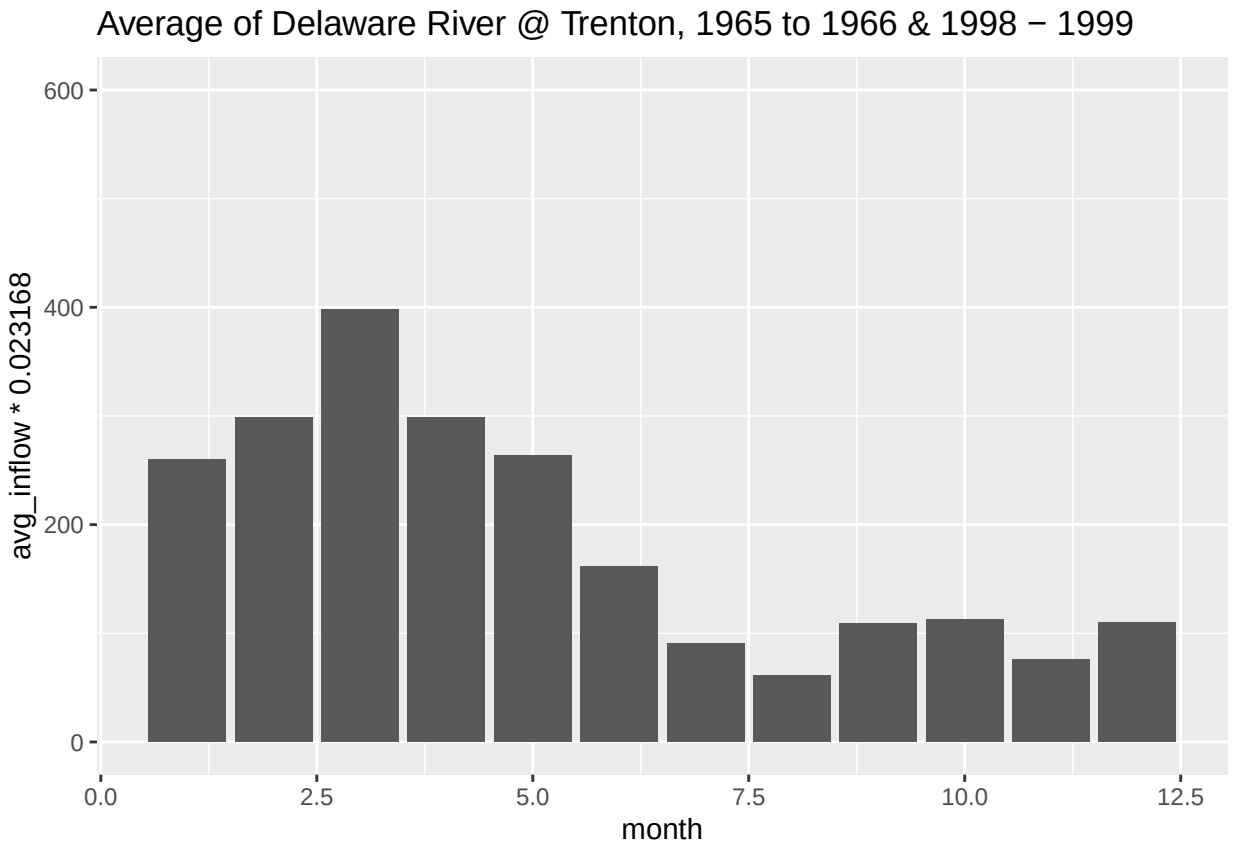
```
colnames(dr) <- c('agency_cd', 'site_no',   'date','missing_4', 'missing_5', 'm6', 'm7', 'm8', 'm9', 'm

dr_clean = as.data.table(dr[, c('agency_cd', 'site_no', 'date', 'avg_discharge_cfps')])
head(dr_clean)
```

```
##    agency_cd  site_no       date avg_discharge_cfps
## 1:      USGS 01463500 1950-08-11               4600
## 2:      USGS 01463500 1950-08-12               4680
## 3:      USGS 01463500 1950-08-13               4320
## 4:      USGS 01463500 1950-08-14               3880
## 5:      USGS 01463500 1950-08-15               3160
## 6:      USGS 01463500 1950-08-16               2960
```

```
dr_clean[, month := month(date)]
ts_dr_flows = dr_clean %>% group_by(month) %>%
  summarise(avg_inflow = mean(avg_discharge_cfps))

ggplot(ts_dr_flows, aes(x = month, y = avg_inflow*.023168))+
  geom_bar(stat = 'identity', position = 'dodge')+
  ylim(0,600)+
  ggtitle('Average of Delaware River @ Trenton, 1950 - 2020')
```




```
ts_dr_flows = dr_clean[(date <= '1999-11-01' & date >= '1997-11-01' ) |
                                (date >= '1964-11-01' & date <= '1966-11-01')] %>% group_by(month) %>%
  summarise(avg_inflow = mean(avg_discharge_cfps))

ggplot(ts_dr_flows, aes(x = month, y = avg_inflow*.023168))+
  geom_bar(stat = 'identity', position = 'dodge')+
  ylim(0,600)+
  ggtitle('Average of Delaware River @ Trenton, 1965 to 1966 & 1998 - 1999')
```



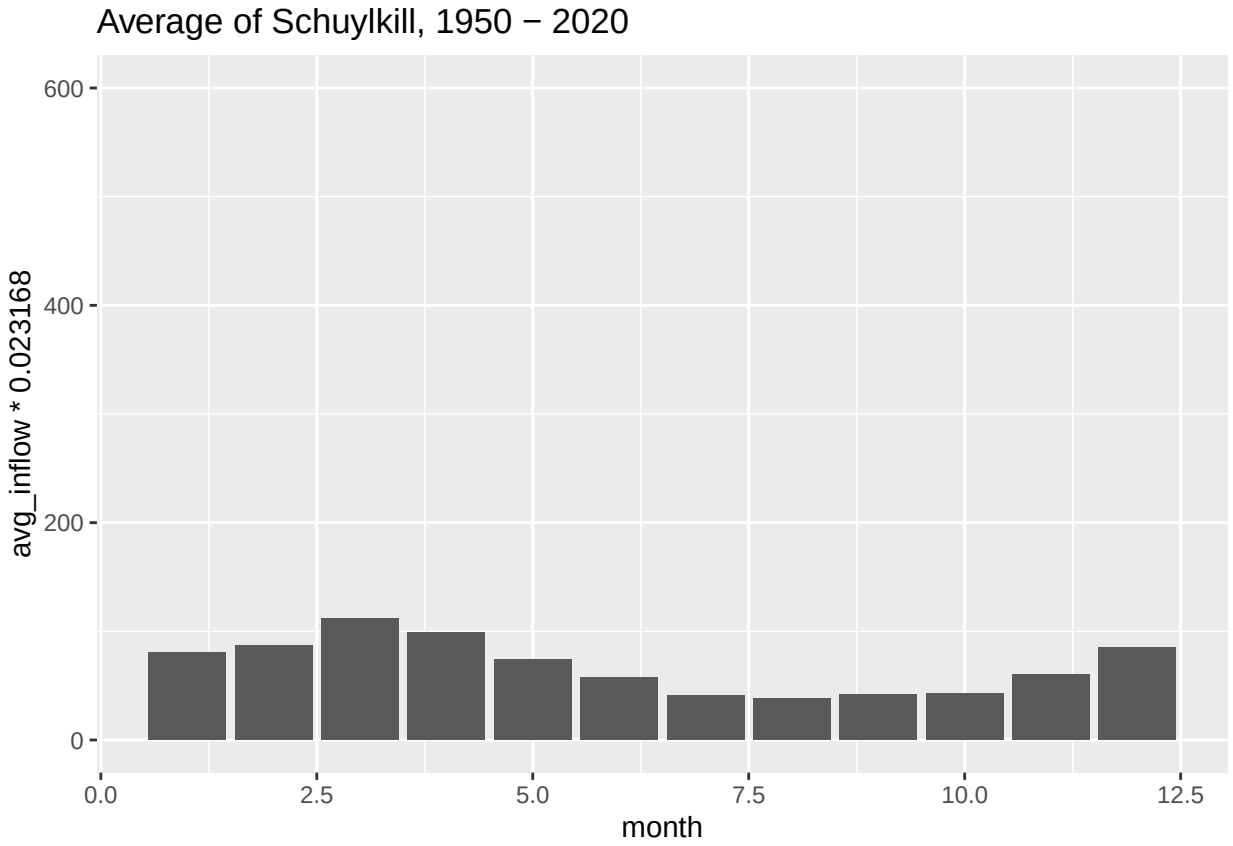
Neither of these plots show a similar seasonal pattern to what is shown in the Meyer paper.

3. Schuylkill River

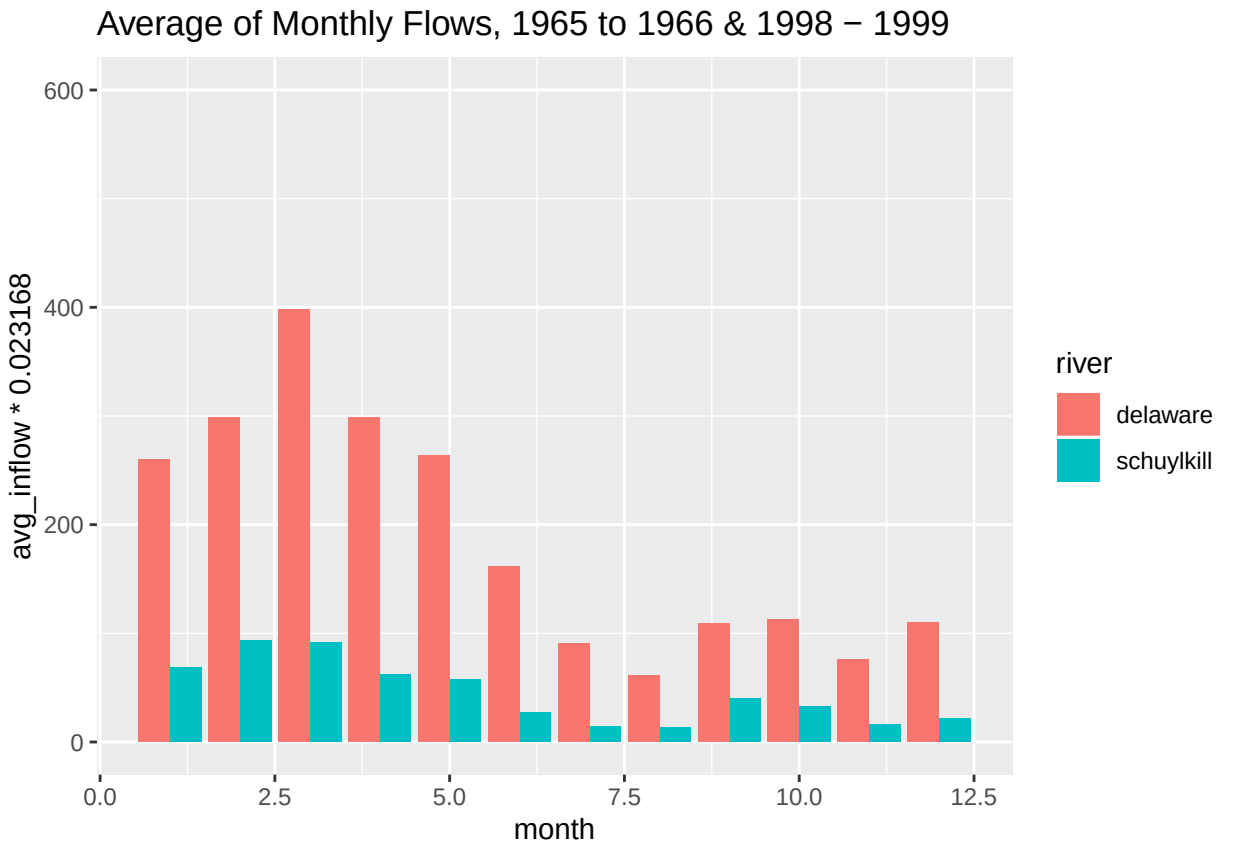
Repeat the process from the Delaware river on the schuylkill river data. This

```
sr <-  read_delim("../data/raw/skill_riv.txt", delim = "\t", skip = 50, col_names = FALSE)

missingness = percent_missing(sr)
# only first 5 columns have have complete data
```

```
colnames(sr) <- c('agency_cd', 'site_no',   'date',  'avg_discharge_cfps', 'code', 'max_sc_mc_sie_p_cm_
sr_clean = as.data.table(sr[, c('agency_cd', 'site_no', 'date',  'avg_discharge_cfps', 'code')])
sr_clean[, month := month(date)]

# our peak raes are march, theirs are in june
# possible difference in data aggregation?
ts_sr_flows = sr_clean %>% group_by(month) %>%
  summarise(avg_inflow = mean(avg_discharge_cfps))

ggplot(ts_sr_flows, aes(x = month, y = avg_inflow*.023168))+
  geom_bar(stat = 'identity', position = 'dodge')+
  ylim(0,600)+
  ggtitle('Average of Schuylkill, 1950 - 2020')
```




```
# plotting inflows only during periods of interest
ts_dr_flows = dr_clean[(date <= '1999-11-01' & date >= '1997-11-01' ) | (date >= '1964-11-01' & date <=

ts_sr_flows = sr_clean[(date <= '1999-11-01' & date >= '1997-11-01' ) | (date >= '1964-11-01' & date <=

#ts_dr_flows = dr_clean %>% group_by(month) %>% summarise(avg_inflow = mean(avg_discharge_cfps))
#ts_sr_flows = sr_clean %>% group_by(month) %>% summarise(avg_inflow = mean(avg_discharge_cfps))

ts_sr_flows = ts_sr_flows %>% mutate(river = 'schuylkill')
ts_dr_flows = ts_dr_flows %>% mutate(river = 'delaware')

flows = rbind(ts_sr_flows, ts_dr_flows)



ggplot(flows, aes(x = month, y = avg_inflow*.023168, fill = river))+
  geom_bar(stat = 'identity', position = 'dodge')+
  ylim(0,600)+
 ggtitle('Average of Monthly Flows, 1965 to 1966 & 1998 - 1999')
```




```
#ggtitle('Average of Monthly Flows, 1950 - 2020')
```

This plot essentially recreates Figure 2 (minus PST model inflows). The ratio of inflow at Delaware vs Schuylkill is consistent with what we see in the Meyer paper, but our data looks more like an inverse bell curve than the normal curve we were expecting?

Next Steps: Ideally we want to figure out why our inflow data doesn't match with what is shown in Figure 2 of the Meyer paper.

SC DATA

4. Chester

```
chester <-  read_delim("../data/raw/chester_sc.txt", delim = "\t", skip = 50, col_names = FALSE)
missingness = percent_missing(chester)
```

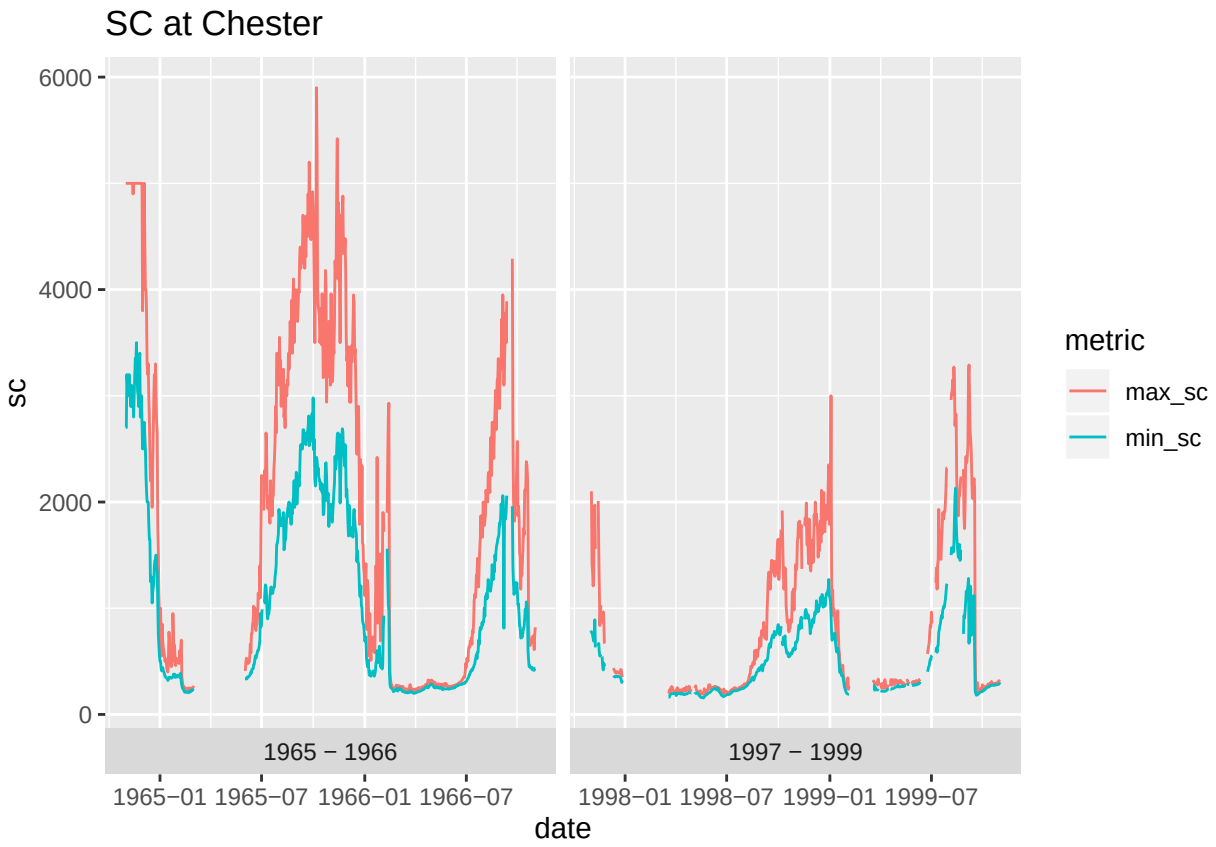
According to the codes I see in the txt file, X4 corresponds to max SC; X6 is min SC, X8 is average SC but it is all missing

```
colnames(chester) <- c('agency_cd', 'site_no',  'date',  'max_sc', 'code', 'min_sc', 'na2', 'min_sc_mc_
chester_clean = as.data.table(chester[, c('agency_cd', 'site_no',   'date',  'max_sc', 'min_sc')])
head(chester_clean)
```

```
##    agency_cd  site_no       date max_sc min_sc
## 1:      USGS 01477050 1961-12-22     NA     NA
## 2:      USGS 01477050 1961-12-23     NA     NA
## 3:      USGS 01477050 1961-12-24     NA     NA
## 4:      USGS 01477050 1961-12-25     NA     NA
```

```
## 5:      USGS 01477050 1961-12-26     NA     NA
## 6:      USGS 01477050 1961-12-27     NA     NA
```

```
chester_clean = as.data.table(chester_clean)[, time_range := ifelse( (date <= '1999-11-01' & date >= '19
                                                                     ifelse( (date >= '1964-11-01' & date <= '19


chester_melt = melt(data = chester_clean, id.vars = c("agency_cd", "site_no", 'date', 'time_range'), mea
setnames(chester_melt, 'variable', 'metric')
setnames(chester_melt, 'value', 'sc')

ggplot(chester_melt[time_range != 'non-relevant'], aes(x = date, y = sc, color = metric))+
  facet_wrap(~time_range, scales = "free_x", switch = 'x')+
  geom_line()+
  ggtitle('SC at Chester')
```



No average SC data is available for the Chester location, but eyeballing the trend it does appear to fall in line with the time series plot shown in the meyer paper.

5. Reedy Island

```
reedy <-  read_delim("../data/raw/reedy_island_sc.txt", delim = "\t", skip = 50, col_names = FALSE)
missingness = percent_missing(reedy)
```

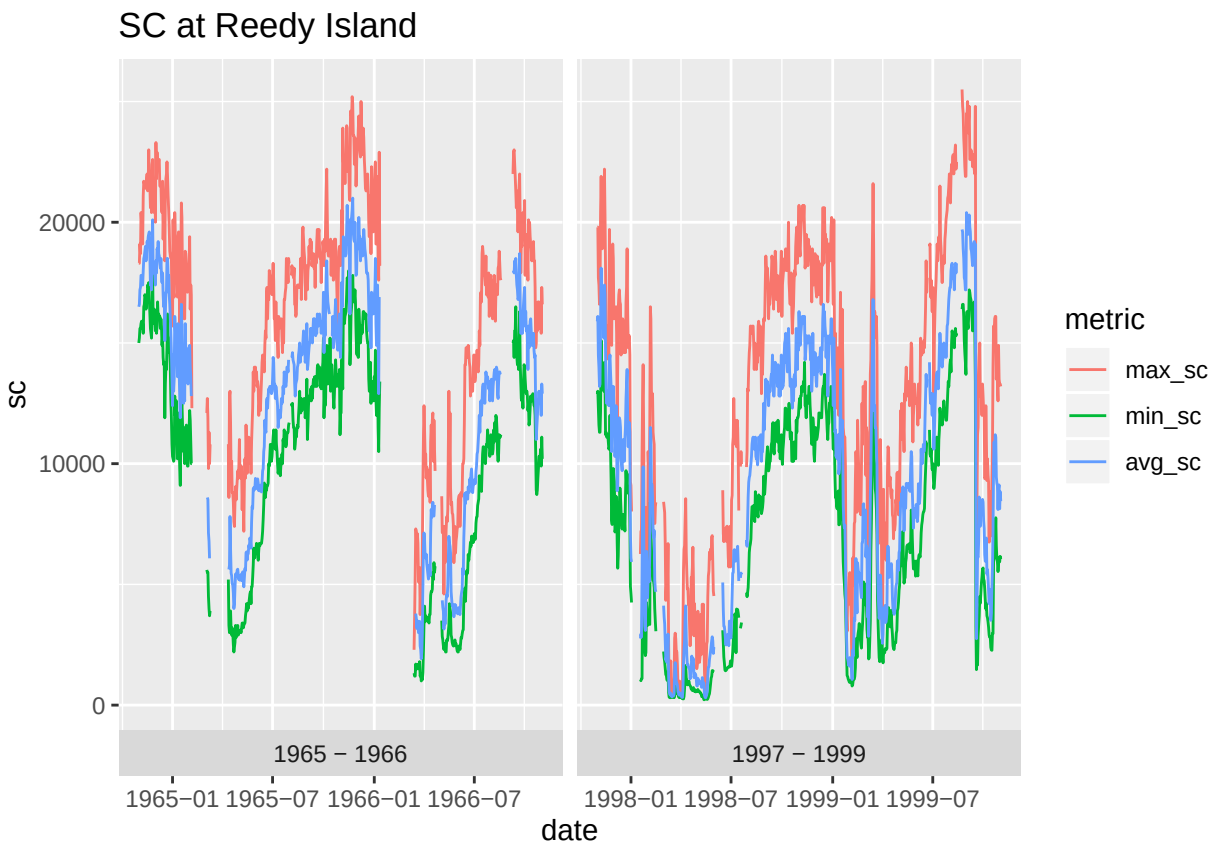
Column 4 is max SC; column 6 is min SC; column 8 is average SC

```
colnames(reedy) <- c('agency_cd', 'site_no',    'date',  'max_sc', 'code', 'min_sc', 'code', 'avg_sc',
reedy_clean = as.data.table(reedy[, c('agency_cd', 'site_no',   'date',  'max_sc', 'min_sc', 'avg_sc')]
```

```
head(reedy_clean)
```

```
##    agency_cd  site_no       date max_sc min_sc avg_sc
## 1:      USGS 01482800 1963-10-07  19500   4370     NA
## 2:      USGS 01482800 1963-10-08  19100   4490     NA
## 3:      USGS 01482800 1963-10-09  20400   4510     NA
## 4:      USGS 01482800 1963-10-10  20500   4700     NA
## 5:      USGS 01482800 1963-10-11  20400   4900     NA
## 6:      USGS 01482800 1963-10-12  19400   4740     NA
```

```
reedy_clean = as.data.table(reedy)[, time_range := ifelse( (date <= '1999-11-01' & date >= '1997-11-01'
                                                            ifelse( (date >= '1964-11-01' & date <= '19



reedy_melt = melt(data = reedy_clean, id.vars = c("agency_cd", "site_no", 'date', 'time_range'), measure
setnames(reedy_melt, 'variable', 'metric')
setnames(reedy_melt, 'value', 'sc')

ggplot(reedy_melt[time_range != 'non-relevant'], aes(x = date, y = sc, color = metric))+
  facet_wrap(~time_range, scales = "free_x", switch = 'x')+
  geom_line()+
  ggtitle('SC at Reedy Island')
```



This seems like a reasonable comparison to what we see in the Meyer paper.

6. Ben Franklin Bridge – missing SC Data?

```
bfb <-  read_delim("../data/raw/ben_franklin_bridge_sc.txt", delim = "\t", skip = 50, col_names = FALSE)
missingness = percent_missing(bfb)
head(missingness)
```

```
## # A tibble: 1 x 41
##      X1    X2    X3    X4    X5    X6    X7    X8    X9   X10   X11   X12   X13
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     0     0     0   100   100   100   100   100   100   100   100   100   100
## # ... with 28 more variables: X14 <dbl>, X15 <dbl>, X16 <dbl>, X17 <dbl>,
## #   X18 <dbl>, X19 <dbl>, X20 <dbl>, X21 <dbl>, X22 <dbl>, X23 <dbl>,
## #   X24 <dbl>, X25 <dbl>, X26 <dbl>, X27 <dbl>, X28 <dbl>, X29 <dbl>,
## #   X30 <dbl>, X31 <dbl>, X32 <dbl>, X33 <dbl>, X34 <dbl>, X35 <dbl>,
## #   X36 <dbl>, X37 <dbl>, X38 <dbl>, X39 <dbl>, X40 <dbl>, X41 <dbl>
```

Columns 1,2,3, 18, 20, 24, 26 have data…But none of those data correspond with SC values (instead they correspond with temperature and dissolved oxygen values). We may need to revisit the data pull in order to get full SC coverage relative to what we see in the Meyer paper.

```
colnames(bfb) <- c('agency_cd', 'site_no',  'date',  'avg_discharge_cfps', 'code', 'max_sc_mc_sie_p_cm_
```