

ADC_Code

Sarah Ko

April 7, 2019

Set up your system

```
# load packages
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.2
## -- Attaching packages ----- t
## v ggplot2 3.1.1    v readr   1.3.1
## v tibble  2.1.1    v purrr  0.3.2
## v tidyr   0.8.3    v stringr 1.4.0
## v ggplot2 3.1.1    v forcats 0.4.0
## Warning: package 'tibble' was built under R version 3.5.3
## Warning: package 'tidyr' was built under R version 3.5.2
## Warning: package 'readr' was built under R version 3.5.2
## Warning: package 'purrr' was built under R version 3.5.3
## Warning: package 'stringr' was built under R version 3.5.2
## Warning: package 'forcats' was built under R version 3.5.2
## -- Conflicts ----- tidyver
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(FSA)
```

```
## Warning: package 'FSA' was built under R version 3.5.3
## ## FSA v0.8.22. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.
```

```
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 3.5.3
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
## The following object is masked from 'package:tidyr':
##
##     expand
library(trend)

## Warning: package 'trend' was built under R version 3.5.2
# set working directory
setwd("~/Duke/Year 2/Spring 2019/Data Analytics/ADC_Analysis/Code")

#check wd
getwd()

## [1] "C:/Users/Sarah/Documents/Duke/Year 2/Spring 2019/Data Analytics/ADC_Analysis/Code"
# create ggplot theme
SKotheme <- theme_gray(base_size = 15) +
  theme(axis.text = element_text(color = "black"),
        legend.position = "right",
        plot.title = element_text(hjust = 0.5))

# set ggplot theme
theme_set(SKotheme)
```

Import & Explore

```
# import dataset
ADC_raw <- read.csv("../Raw_Data/CalRecycle_ADC_raw.csv")

# explore dataset
view(ADC_raw)
class(ADC_raw)

## [1] "data.frame"
colnames(ADC_raw)

## [1] "Report.Year"
## [2] "Report.Quarter"
## [3] "Ash"
## [4] "Auto.Shredder.Waste"
## [5] "Construction.and.Demolition.Waste"
## [6] "Compost"
## [7] "Contaminated.Sediment"
## [8] "Green.Material"
## [9] "Mixed"
## [10] "Other"
## [11] "Tires"
## [12] "Sludge"
## [13] "Total"
```

```

dim(ADC_raw)

## [1] 92 13

# per the CalRecycle website, segregation into ADC types started in 1998
# therefore, for the analysis, remove data from before 1998
class(ADC_raw$Report.Year)

## [1] "integer"

ADC_data <- filter(ADC_raw, Report.Year >= 1998)
dim(ADC_data)

## [1] 80 13

# explore new dataset
head(ADC_data)

##   Report.Year Report.Quarter      Ash Auto.Shredder.Waste
## 1         2017              1 32511.83          153270.6
## 2         2017              2 37294.78          159759.7
## 3         2017              3 33349.25          153342.6
## 4         2017              4 22248.85          123203.5
## 5         2016              1 31423.40          123193.3
## 6         2016              2 45504.45          126040.9
##   Construction.and.Demolition.Waste Compost Contaminated.Sediment
## 1                               173548.6 6128.89          3396.36
## 2                               199486.8 2746.22          7585.58
## 3                               164028.4 1796.97          4280.92
## 4                               198901.7 15993.13         2979.12
## 5                               160446.5 15681.63        20203.18
## 6                               144982.9 42215.62        18089.73
##   Green.Material   Mixed   Other   Tires   Sludge   Total
## 1    380686.2      0.00 71983.68 3771.40 68063.34 893360.9
## 2    401034.3    1516.12 71066.46 5066.35 65585.25 951141.6
## 3    362474.4   10891.73 78980.55 5323.75 79967.05 894435.6
## 4    347204.0    7964.83 56849.63 4575.75 141423.92 921344.5
## 5    334512.7   12756.90 82081.97 3402.03 83424.85 867126.5
## 6    310959.5   17946.71 75803.52 3616.26 72882.61 858042.2

tail(ADC_data)

##   Report.Year Report.Quarter      Ash Auto.Shredder.Waste
## 75         1999              3 1578.70          69300.25
## 76         1999              4 2718.22          63910.19
## 77         1998              1 2631.85          39181.17
## 78         1998              2  878.63          49391.25
## 79         1998              3 2457.00          35573.00
## 80         1998              4 2418.00          38495.89
##   Construction.and.Demolition.Waste Compost Contaminated.Sediment
## 75                               48321.13      0          0.00
## 76                               62057.02    381         16.50
## 77                               2693.48      0          0.00
## 78                               6666.70      0          2.74
## 79                               28278.30      0         92.17
## 80                               29591.80      0         0.00
##   Green.Material   Mixed   Other   Tires   Sludge   Total

```

```
## 75      349276.6      0.00 4695.69  1265.82 66864.38 541302.6
## 76      360153.2      0.00 6316.72  3307.48 69058.27 567918.6
## 77      191066.3 3907.20 1008.27 14802.71 43391.12 298682.1
## 78      279191.3 3602.22 3305.93 15394.54 92416.47 450849.8
## 79      299986.8      0.00 2706.53  2943.31 99312.34 471349.4
## 80      313452.3 4130.00 3767.93   733.71 57511.25 450100.9

# tidy the data by gathering the type columns
ADC_gathered <- gather(ADC_data, "Type", "Quantity", Ash:Sludge) %>%
  select(-Total) # remove Total column

# save the tidy dataset
write.csv(ADC_data, row.names = FALSE, file = "../Processed_Data/CalRecycle_ADC_tidy_processed.csv")

# generate summary data
ADC_summary_by_type <- ADC_gathered %>%
  group_by(Type) %>% # group the data by lakenname
  filter(!is.na(Quantity)) %>% #remove the records when there are nas Quantity
  summarise(MeanQuarterlyQuantity = mean(Quantity),
            MinQuarterlyQuantity = min(Quantity),
            MaxQuarterlyQuantity = max(Quantity),
            sdQuarterlyQuantity = sd(Quantity),
            medianQuarterlyQuantity = median(Quantity))

ADC_summary_by_year <- ADC_gathered %>%
  group_by(Report.Year) %>% # group the data by year
  filter(!is.na(Quantity)) %>% #remove the records when there are nas Quantity
  summarise(MeanQuarterlyQuantity = mean(Quantity),
            MinQuarterlyQuantity = min(Quantity),
            MaxQuarterlyQuantity = max(Quantity),
            sdQuarterlyQuantity = sd(Quantity),
            medianQuarterlyQuantity = median(Quantity))
```

Create Graphs

```
# Graph 1: for 2017 data, display total by type
total_bytype_2017 <- ADC_gathered %>%
  filter(Report.Year == 2017) %>%
  group_by(Type) %>%
  summarize(Quantity = sum(Quantity))

# save 2017 dataset
write.csv(total_bytype_2017, row.names = FALSE, file = "../Processed_Data/CalRecycle_ADC_2017only_processed.csv")

# convert column Type into factor
class(total_bytype_2017$Type)

## [1] "character"

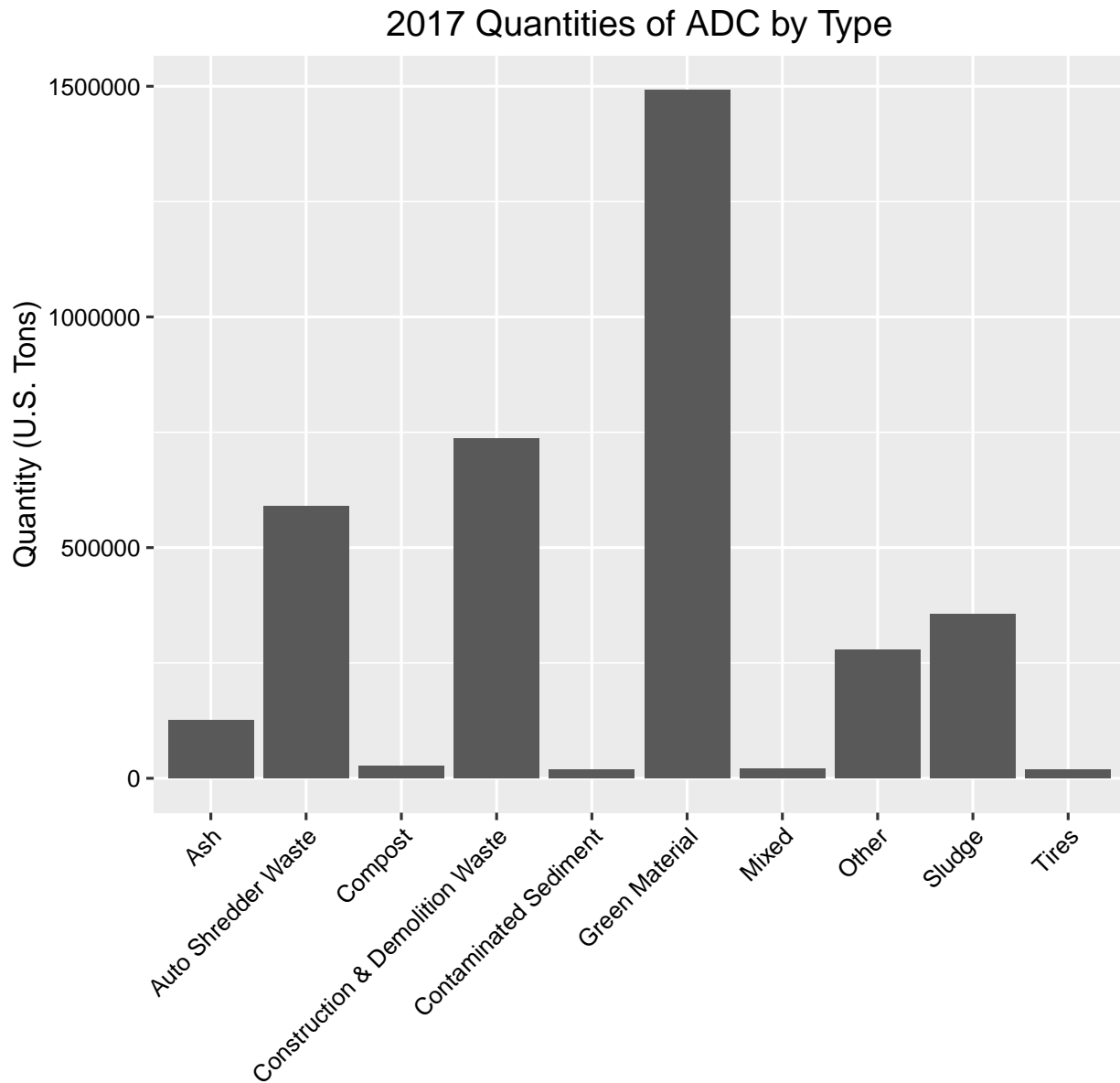
total_bytype_2017$Type <- as.factor(total_bytype_2017$Type)

# plot as a bar chart
total_bytype_2017_plot <-
  ggplot(data=total_bytype_2017, aes(x=Type, y=Quantity)) +
  geom_bar(stat="identity") +
```

```

xlab('') +
ylab("Quantity (U.S. Tons)") +
ggtitle("2017 Quantities of ADC by Type") +
theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
scale_x_discrete(labels = c('Ash', 'Auto Shredder Waste', 'Compost', 'Construction & Demolition Waste', 'Contaminated Sediment', 'Green Material', 'Mixed', 'Other', 'Sludge', 'Tires'))
print(total_bytype_2017_plot)

```



```

# save figure
ggsave("2017ADCbytype_alltypes.jpg", total_bytype_2017_plot, path = "../Output", height = 4, width = 6,

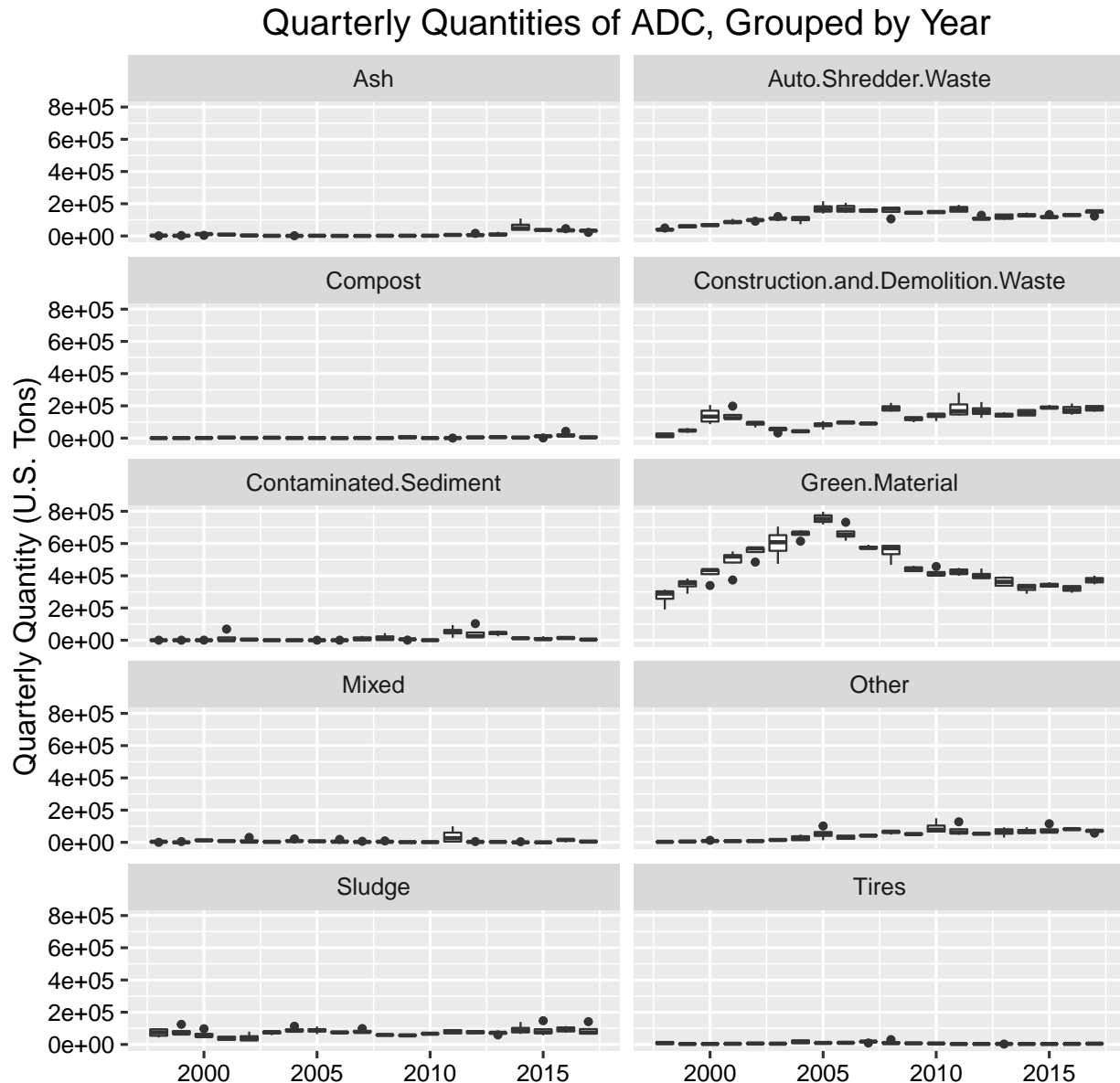
# Graph 2: faceted by Type, display spread of quarterly values by year
quarterlyvalues_byyear_plot <- ggplot(ADC_gathered) +
  geom_boxplot(aes(x = Report.Year, y = Quantity, group = Report.Year)) +
  facet_wrap(vars(Type), nrow = 5) +

```

```

xlab("") +
ylab("Quarterly Quantity (U.S. Tons)") +
ggtitle("Quarterly Quantities of ADC, Grouped by Year")
print(quarterlyvalues_byyear_plot)

```



```

# save figure
ggsave("ADCyeardistribution_alltypes.jpg", quarterlyvalues_byyear_plot, path = "../Output", height = 8,

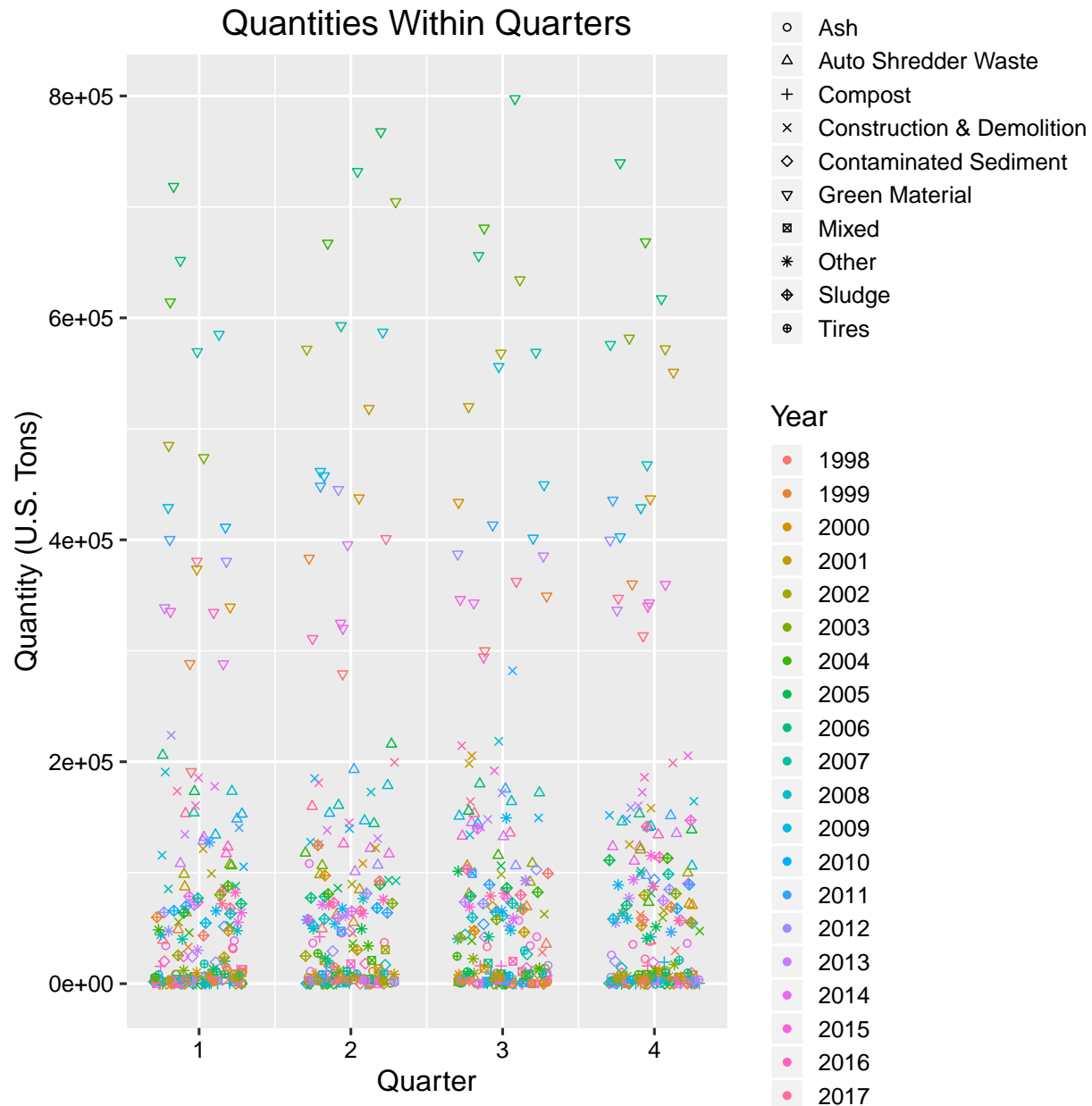
# Graph 3: display data by quarter, all Types on same plot
quarterlyvalues_alltypes_plot <-
  ggplot(ADC_gathered) +
  geom_jitter(aes(x = Report.Quarter, y = Quantity, shape = as.factor(Type), color = as.factor(Report.Y
  labs(shape="Type", colour="Year") +
  xlab("Quarter") +

```

```

ylab("Quantity (U.S. Tons)") +
ggtitle("Quantities Within Quarters") +
scale_shape_manual(values=c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10), labels = c("Ash", "Auto Shredder Waste",
theme(legend.position="right", legend.box = "vertical", legend.direction = "vertical") +
guides(shape = guide_legend(order = 1), color = guide_legend(order = 2))
print(quarterlyvalues_alltypes_plot)

```



```

# save figure
ggsave("QuarterlyADC_alltypes.jpg", quarterlyvalues_alltypes_plot, path = "../Output", height = 9, width

```

Test 1: Statistical Modeling & Data Visualization

Is there a significant difference in total ADC between report quarters? (e.g. 1, 2, 3, 4)

```

# create dataset with only total values, from 1995-2017
ADC_total_only <- ADC_raw %>%
  select(Report.Year, Report.Quarter, Total) # keep all columns except ADC Types

# convert column Report.Quarter into factor
class(ADC_total_only$Report.Quarter)

## [1] "integer"
ADC_total_only$Report.Quarter <- as.factor(ADC_total_only$Report.Quarter)

# save the dataset
write.csv(ADC_total_only, row.names = FALSE, file = "../Processed_Data/CalRecycle_ADC_totalonly_processed.csv")

# perform one-way ANOVA
# assumption #0: observations are independent (cannot be tested, but assumed to be independent)

# test assumption #1: normality
# null hypothesis is that the dataset is normally distributed
shapiro.test(ADC_total_only$Total[ADC_total_only$Report.Quarter == 1]) # p-value = 0.03312

##
##  Shapiro-Wilk normality test
##
## data:  ADC_total_only$Total[ADC_total_only$Report.Quarter == 1]
## W = 0.90566, p-value = 0.03312
shapiro.test(ADC_total_only$Total[ADC_total_only$Report.Quarter == 2]) # p-value = 0.02271

##
##  Shapiro-Wilk normality test
##
## data:  ADC_total_only$Total[ADC_total_only$Report.Quarter == 2]
## W = 0.89774, p-value = 0.02271
shapiro.test(ADC_total_only$Total[ADC_total_only$Report.Quarter == 3]) # p-value = 0.00993

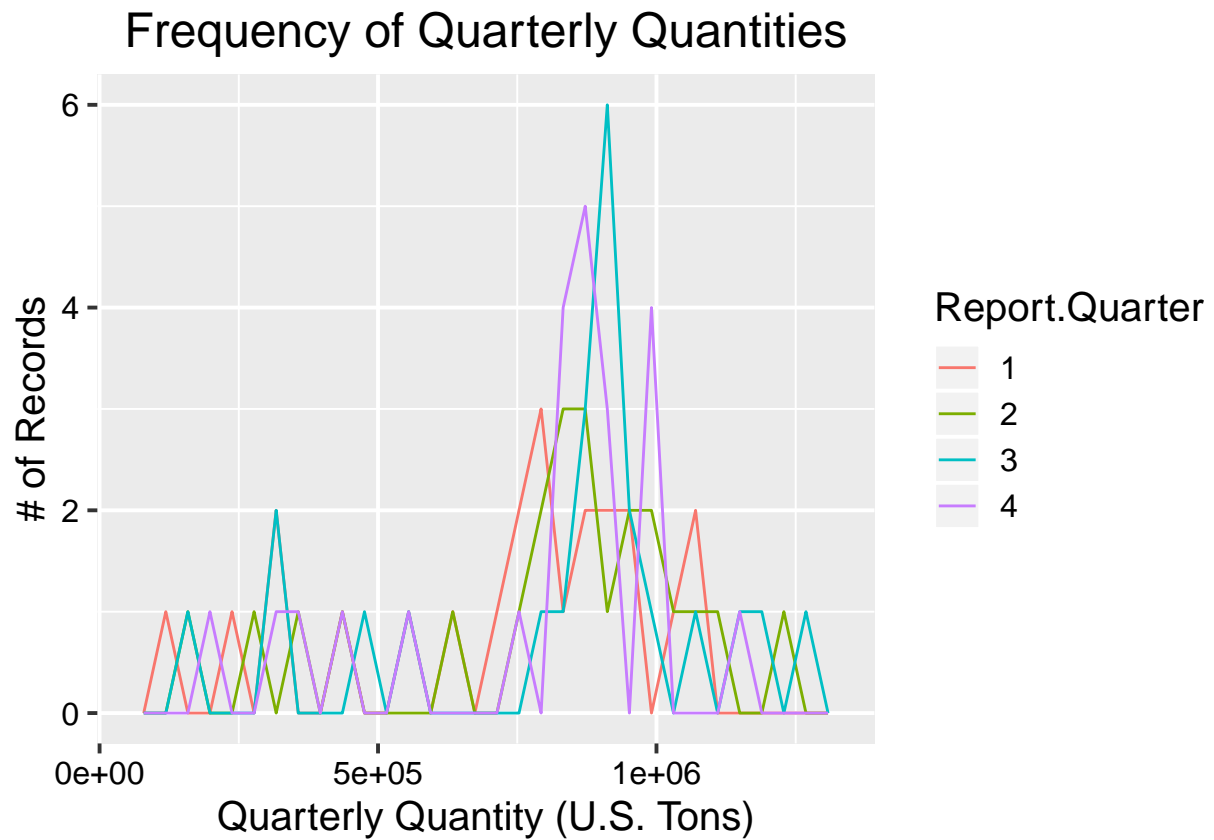
##
##  Shapiro-Wilk normality test
##
## data:  ADC_total_only$Total[ADC_total_only$Report.Quarter == 3]
## W = 0.87982, p-value = 0.00993
shapiro.test(ADC_total_only$Total[ADC_total_only$Report.Quarter == 4]) # p-value = 0.001305

##
##  Shapiro-Wilk normality test
##
## data:  ADC_total_only$Total[ADC_total_only$Report.Quarter == 4]
## W = 0.83198, p-value = 0.001305
ADC_freq_poly <- ggplot(ADC_total_only) +
  geom_freqpoly(aes(x = Total, color = Report.Quarter)) +
  xlab("Quarterly Quantity (U.S. Tons)") +
  ylab("# of Records") +
  ggtitle("Frequency of Quarterly Quantities")
print(ADC_freq_poly) # appears to be left skewed

```

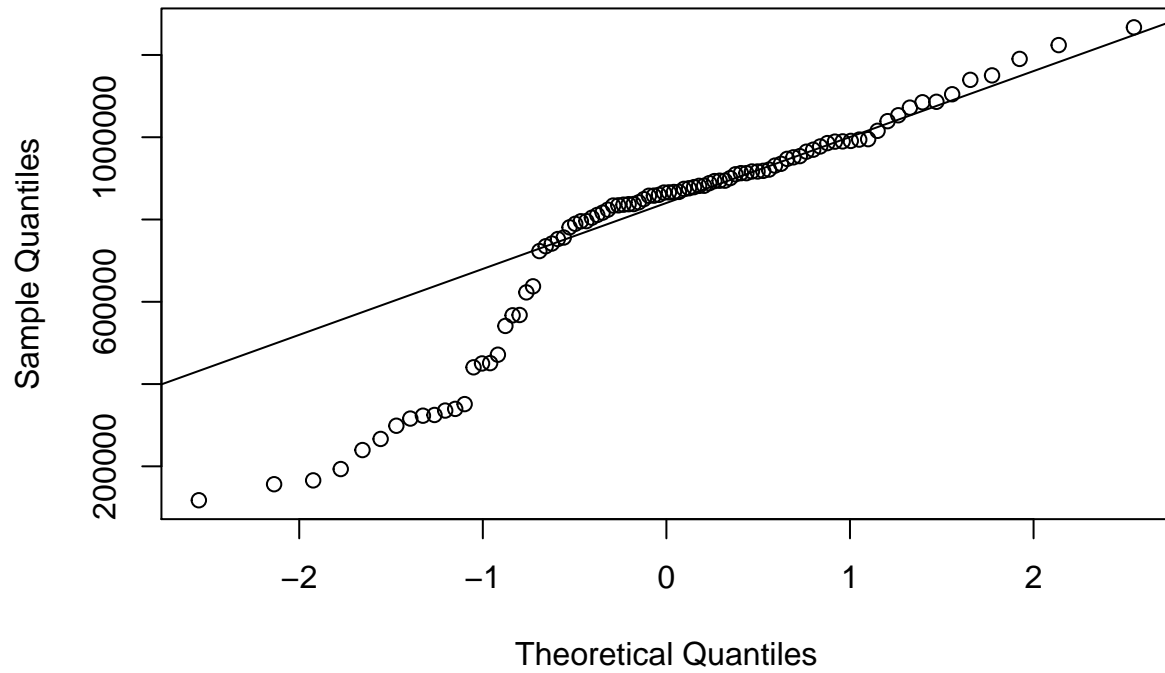


```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



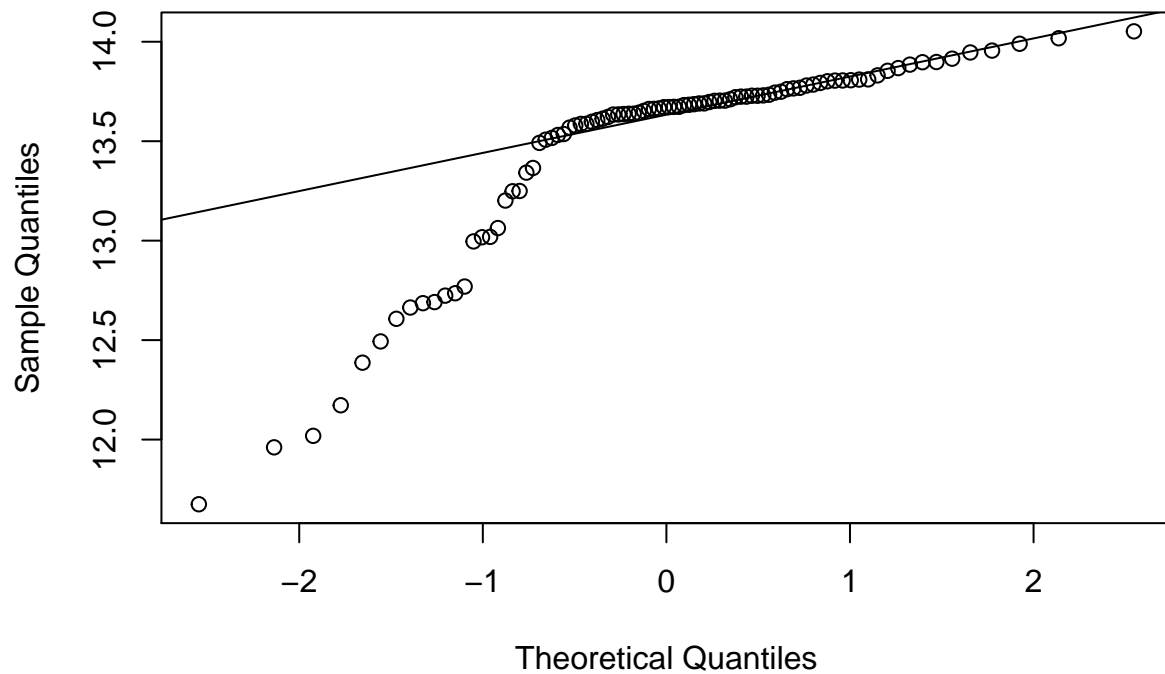
```
qqnorm(ADC_total_only$Total); qqline(ADC_total_only$Total) # does not match 1:1 ratio
```

Normal Q-Q Plot



```
# Try to fix departure from normality with ln of Total. Result is not improved, so keep non-transformed  
ADC_LogTotal <- mutate(ADC_total_only, LogTotal = log(Total))  
qqnorm(ADC_LogTotal$LogTotal); qqline(ADC_LogTotal$LogTotal)
```

Normal Q-Q Plot



```
bartlett.test(ADC_LogTotal$LogTotal ~ ADC_LogTotal$Report.Quarter)
```

```
##
```

```
## Bartlett test of homogeneity of variances
```

```
##
```

```
## data: ADC_LogTotal$LogTotal by ADC_LogTotal$Report.Quarter
```

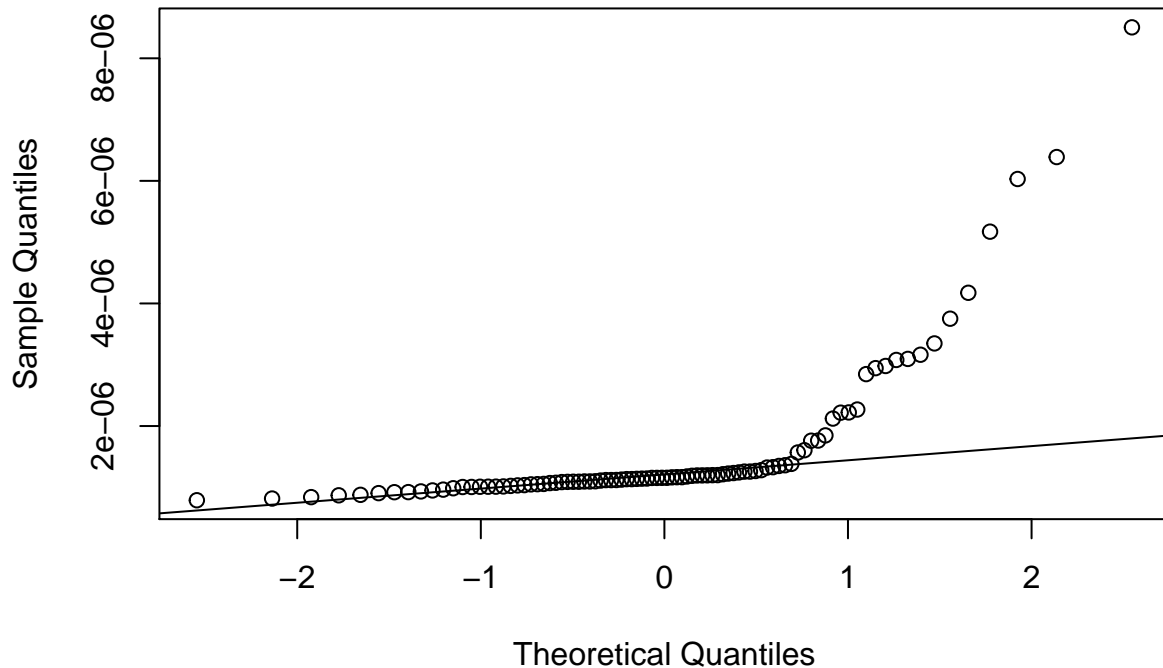
```
## Bartlett's K-squared = 1.1435, df = 3, p-value = 0.7666
```

```
# Try to fix departure from normality with 1/Total. Result is not improved, so keep non-transformed data
```

```
ADC_InvTotal <- mutate(ADC_total_only, InvTotal = 1/Total)
```

```
qqnorm(ADC_InvTotal$InvTotal); qqline(ADC_InvTotal$InvTotal)
```

Normal Q-Q Plot



```
bartlett.test(ADC_InvTotal$InvTotal ~ ADC_InvTotal$Report.Quarter)
```

```
##
## Bartlett test of homogeneity of variances
##
## data: ADC_InvTotal$InvTotal by ADC_InvTotal$Report.Quarter
## Bartlett's K-squared = 6.519, df = 3, p-value = 0.08892
```

```
# test assumption #2: equal variances among groups
```

```
# null hypothesis is that the variance is the same for the treatment groups
```

```
bartlett.test(ADC_total_only$Total ~ ADC_total_only$Report.Quarter) #p-value = 0.9308 # df = 3 (statist
```

```
##
## Bartlett test of homogeneity of variances
##
## data: ADC_total_only$Total by ADC_total_only$Report.Quarter
## Bartlett's K-squared = 0.44478, df = 3, p-value = 0.9308
```

```
# dataset is not normal, but does fulfill requirement for same variances. proceed with non-parametric t
```

```
# try non-parametric w/ post hoc, bc sample size is on the smaller end for parametric
```

```
ADC_quarter_kw <- kruskal.test(ADC_total_only$Total ~ ADC_total_only$Report.Quarter)
ADC_quarter_kw
```

```
##
## Kruskal-Wallis rank sum test
##
```

```
## data: ADC_total_only$Total by ADC_total_only$Report.Quarter
## Kruskal-Wallis chi-squared = 3.4581, df = 3, p-value = 0.3262
```

```
dunnTest(ADC_total_only$Total, ADC_total_only$Report.Quarter)
```

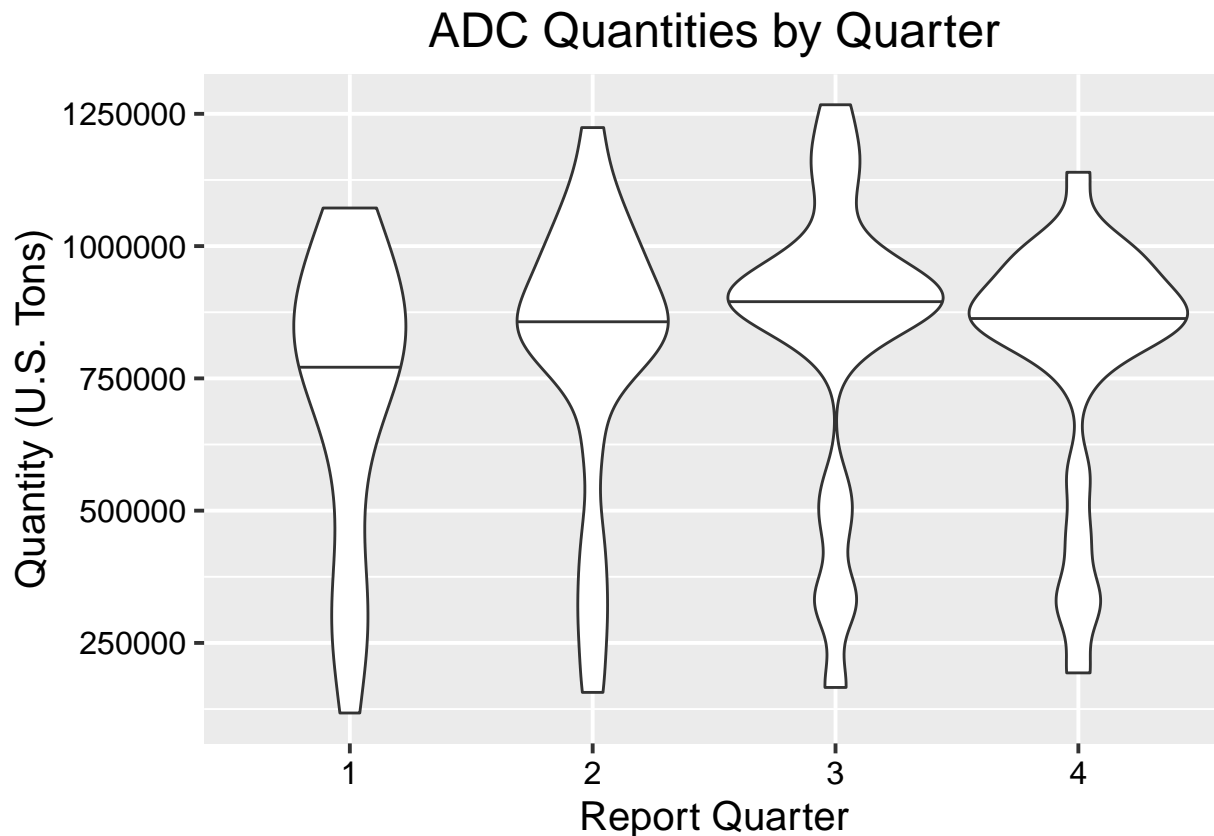
```
## Dunn (1964) Kruskal-Wallis multiple comparison
```

```
## p-values adjusted with the Holm method.
```

```
## Comparison      Z    P.unadj    P.adj
## 1      1 - 2 -1.08778370 0.27669061 1.0000000
## 2      1 - 3 -1.84978446 0.06434462 0.3860677
## 3      2 - 3 -0.76200076 0.44605955 0.8921191
## 4      1 - 4 -1.00495753 0.31491730 1.0000000
## 5      2 - 4  0.08282617 0.93398976 0.9339898
## 6      3 - 4  0.84482693 0.39820748 1.0000000
```

```
# plot the results
```

```
ADC_quarter_plot <- ggplot(ADC_total_only, aes(x = Report.Quarter, y = Total)) +
  geom_violin(draw_quantiles = 0.5) +
  xlab('Report Quarter') +
  ylab('Quantity (U.S. Tons)') +
  ggtitle('ADC Quantities by Quarter')
print(ADC_quarter_plot)
```



```
# save figure
```

```
ggsave("QuarterlyADC_violinplot.jpg", ADC_quarter_plot, path = "../Output", height = 4, width = 6, units = "in")
```

Test 2: Statistical Modeling & Data Visualization

Can total annual ADC be represented with a linear model?

```
# assumptions for lm (independent observation, normal distribution, equal variances among groups) check

# create dates corresponding to year & quarter combination
# Q1: Mar 31
# Q2: Jun 30
# Q3: Sep 30
# Q4: Dec 31

# create dataframe of month-date
quarters_to_dates <- data.frame("Quarter" = as.factor(1:4), "Month.Date" = c('3-31', '6-30', '9-30', '12-31'))

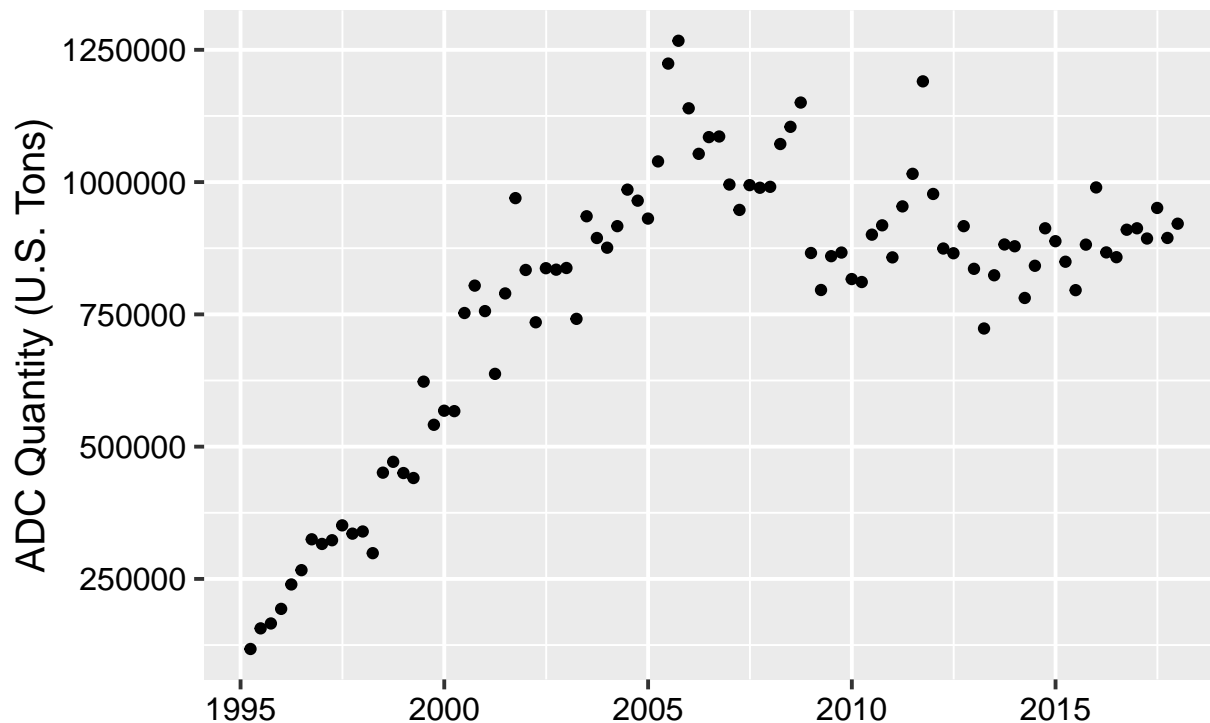
# create new dataframe with dates
ADC_fulldate <- ADC_total_only %>%
  inner_join(quarters_to_dates, by = c("Report.Quarter" = "Quarter")) %>%
  unite('Quarter.End.Date', c(Report.Year, Month.Date), sep = "-", remove = FALSE)

ADC_fulldate$Quarter.End.Date <- as.Date(ADC_fulldate$Quarter.End.Date, "%Y-%m-%d")
class(ADC_fulldate$Quarter.End.Date)

## [1] "Date"

# create initial plot to visualize the data
ggplot(ADC_fulldate, aes(x = Quarter.End.Date, y = Total)) +
  geom_point() +
  xlab("") +
  ylab("ADC Quantity (U.S. Tons)") +
  ggtitle("Quarterly Quantities of ADC")
```

Quarterly Quantities of ADC



```
# create lm
ADC_date_lm <- lm(data = ADC_fulldate, Total ~ Quarter.End.Date)
ADC_date_lm # Total = 73.14*Quarter.End.Date - 190264.58
```

```
##
## Call:
## lm(formula = Total ~ Quarter.End.Date, data = ADC_fulldate)
##
## Coefficients:
## (Intercept) Quarter.End.Date
## -190264.58 73.14
```

```
summary(ADC_date_lm) # Adjusted R-squared: 0.4433 (date explains 44.33% of variation in total), p-value
```

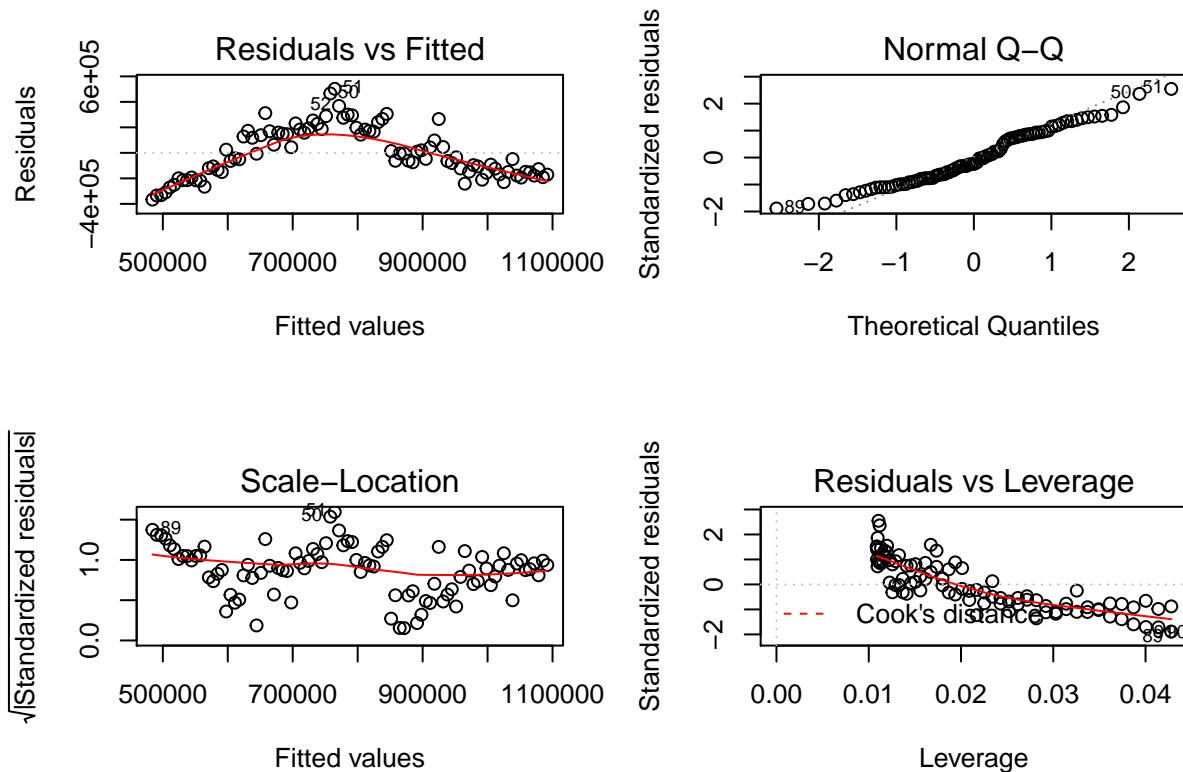
```
##
## Call:
## lm(formula = Total ~ Quarter.End.Date, data = ADC_fulldate)
##
## Residuals:
## Min 1Q Median 3Q Max
## -366483 -153515 -45160 167108 502499
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.903e+05 1.160e+05 -1.64 0.104
## Quarter.End.Date 7.314e+01 8.534e+00 8.57 2.69e-13 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 198500 on 90 degrees of freedom
## Multiple R-squared:  0.4494, Adjusted R-squared:  0.4433
## F-statistic: 73.45 on 1 and 90 DF,  p-value: 2.694e-13
```

```
# check normality of residuals
```

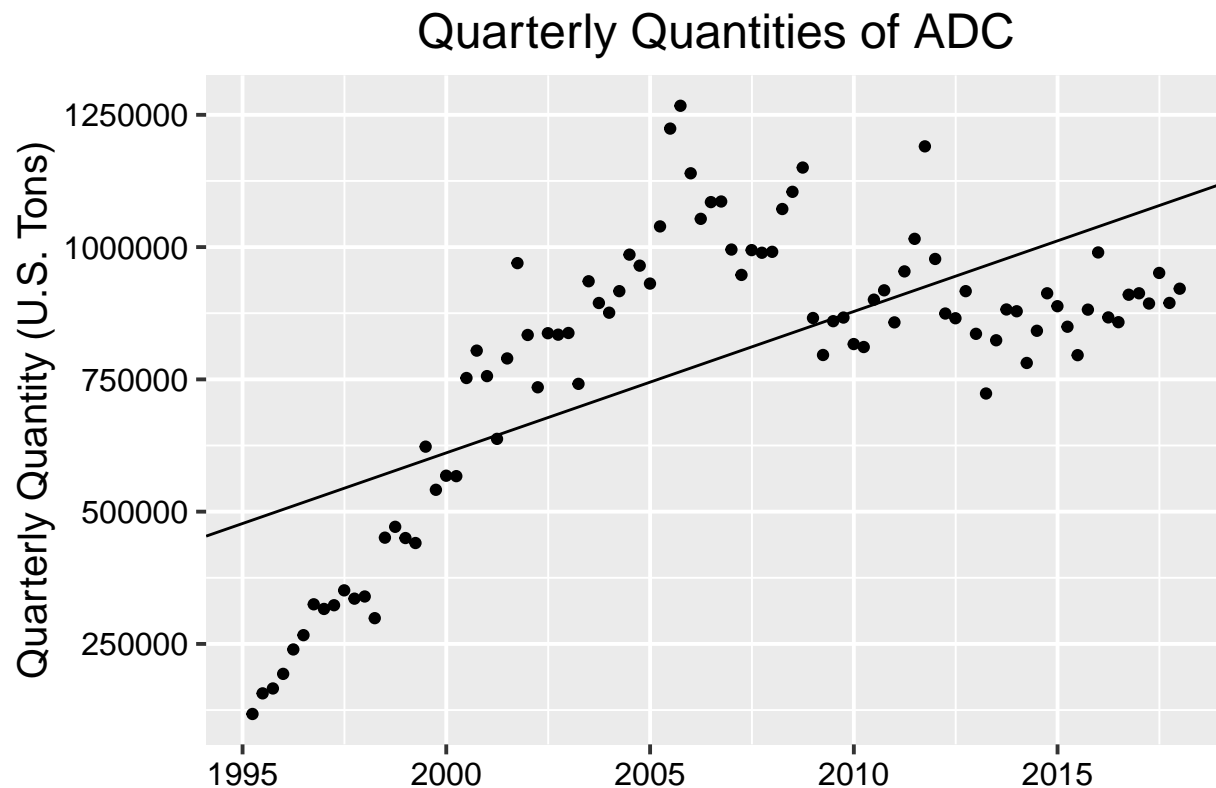
```
par(mfrow=c(2,2))
```

```
plot(ADC_date_lm) # QQ of residuals looks relatively normal
```



```
# plot data w/ model
```

```
ADC_fulldate_plot <- ggplot(ADC_fulldate, aes(x = Quarter.End.Date, y = Total)) +
  geom_abline(intercept = -190264.58, slope = 73.14) +
  geom_point() +
  xlab('') +
  ylab('Quarterly Quantity (U.S. Tons)') +
  ggtitle('Quarterly Quantities of ADC')
print(ADC_fulldate_plot)
```

```
# visually, model does not appear to be a great fit
```

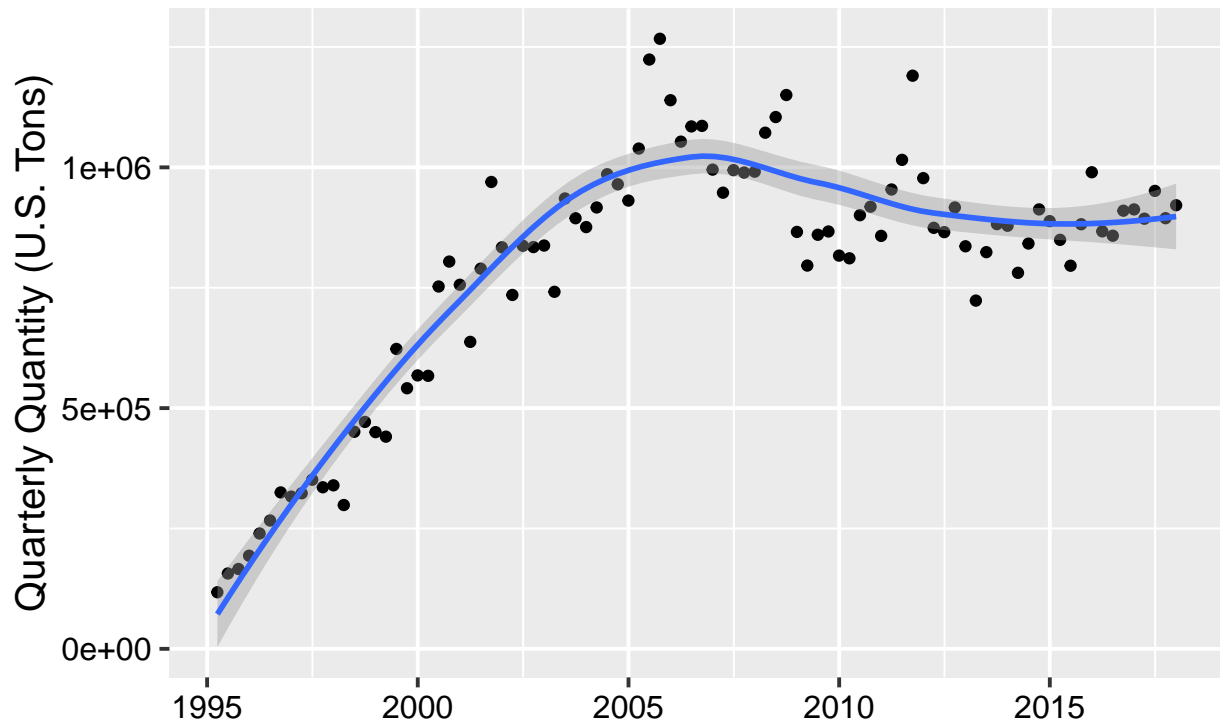
```
# save figure
```

```
ggsave("TotalADC_plot_calculatedmodel.jpg", ADC_fulldate_plot, path = "../Output", height = 4, width = 6)
```

```
# plot with loess smoother
```

```
ADC_fulldate_plot_loess <- ggplot(ADC_fulldate, aes(x = Quarter.End.Date, y = Total)) +  
  geom_point() +  
  geom_smooth(method = loess) +  
  xlab('') +  
  ylab('Quarterly Quantity (U.S. Tons)') +  
  ggtitle('Quarterly Quantities of ADC')  
print(ADC_fulldate_plot_loess)
```

Quarterly Quantities of ADC



```
# visually, model appears to be a great fit
```

```
# save figure
```

```
ggsave("TotalADC_plot_loess.jpg", ADC_fulldate_plot_loess, path = "../Output", height = 4, width = 6, u
```

Test 3: Statistical Modeling & Data Visualization

Is there a changepoint in the Construction & Demolition quantities over time?

```
# create dataframe with dates
```

```
quarters_to_dates$Quarter <- as.integer(quarters_to_dates$Quarter)
```

```
CD_only <- ADC_data %>%
```

```
  select(Report.Year, Report.Quarter, Construction.and.Demolition.Waste) %>%
```

```
  inner_join(quarters_to_dates, by = c("Report.Quarter" = "Quarter")) %>%
```

```
  unite('Quarter.End.Date', c(Report.Year, Month.Date), sep = "-") %>%
```

```
  select(-Report.Quarter)
```

```
CD_only$Quarter.End.Date <- as.Date(CD_only$Quarter.End.Date, '%Y-%m-%d') # format column as date
```

```
# arrange data from oldest to newest
```

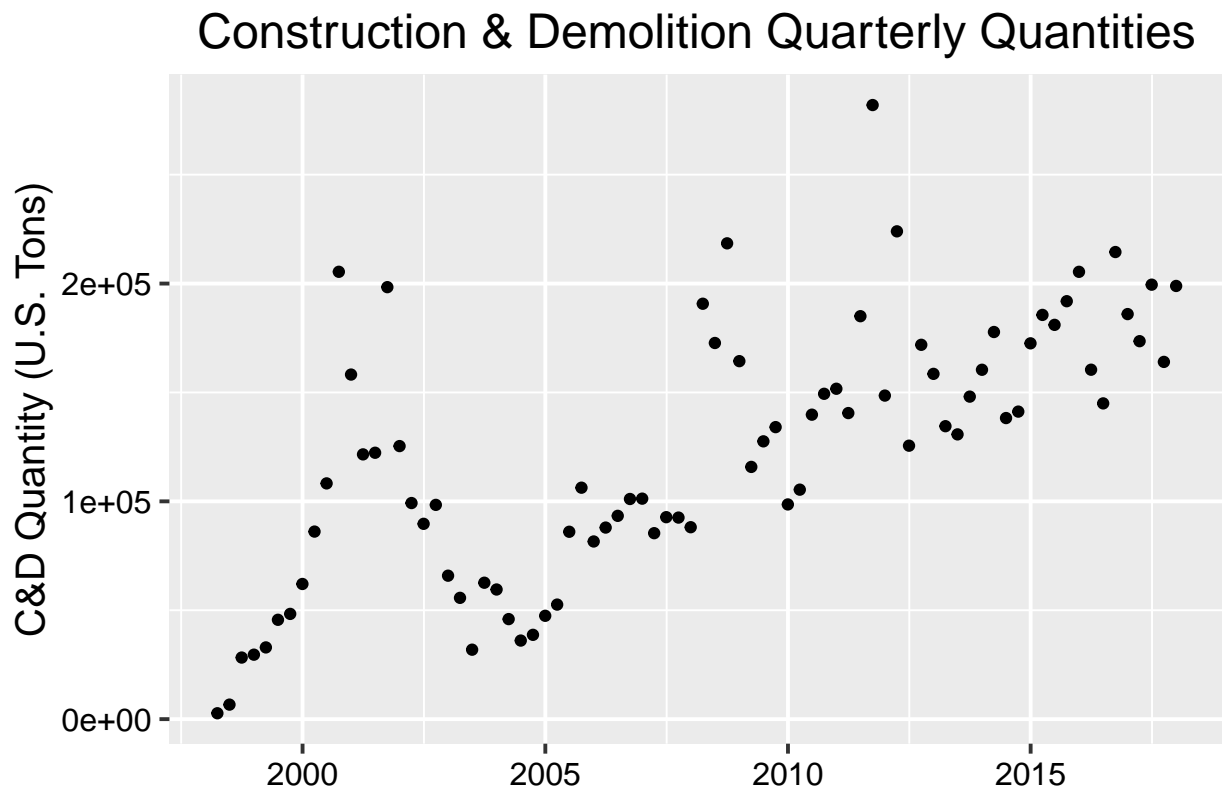
```
CD_only <- CD_only %>%
```

```
  arrange(Quarter.End.Date)
```

```
# create initial plot to visualize the data
```

```
ggplot(CD_only, aes(x = Quarter.End.Date, y = Construction.and.Demolition.Waste)) +  
  geom_point() +
```

```
xlab("") +
ylab("C&D Quantity (U.S. Tons)") +
ggtitle("Construction & Demolition Quarterly Quantities")
```

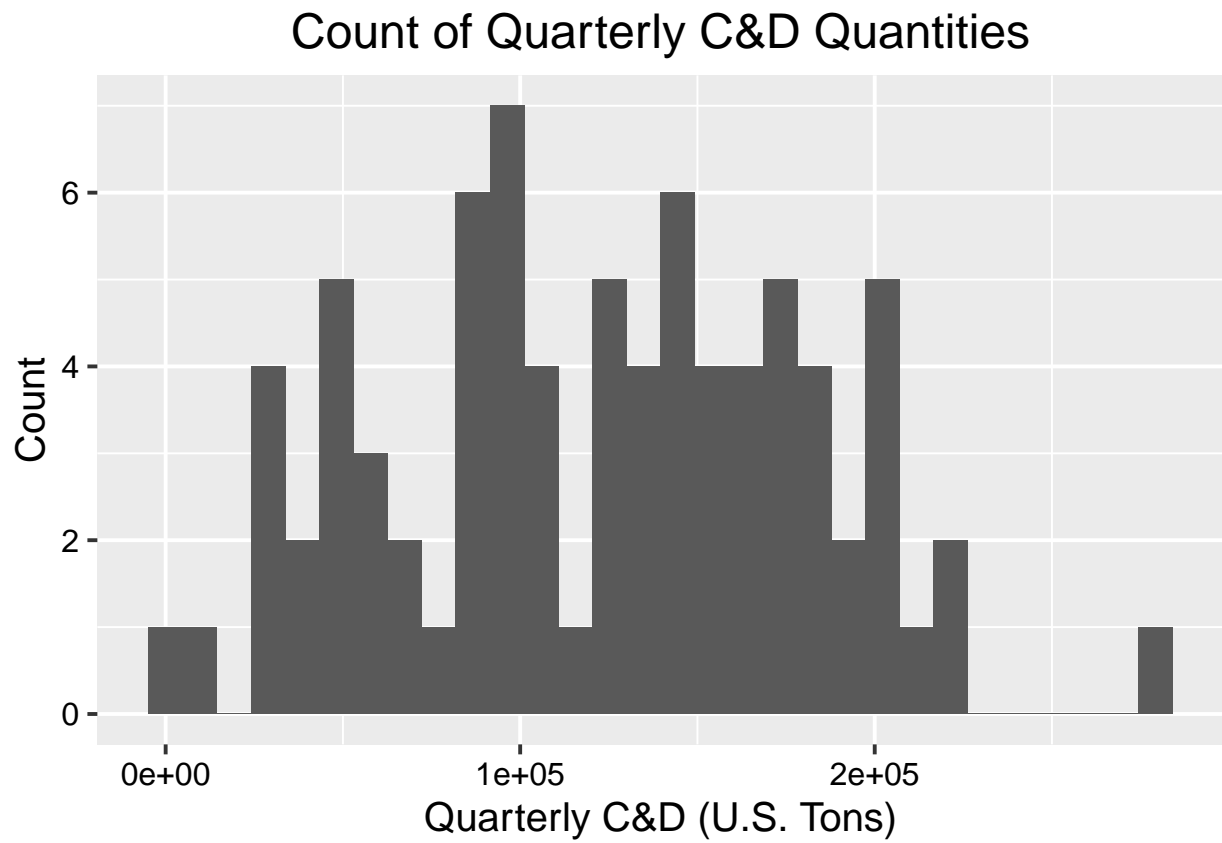


```
# check normality for C&D waste specifically
shapiro.test(CD_only$Construction.and.Demolition.Waste) # p-value = 0.4028, inferring that the data is normal
```

```
##
## Shapiro-Wilk normality test
##
## data: CD_only$Construction.and.Demolition.Waste
## W = 0.9837, p-value = 0.4028
```

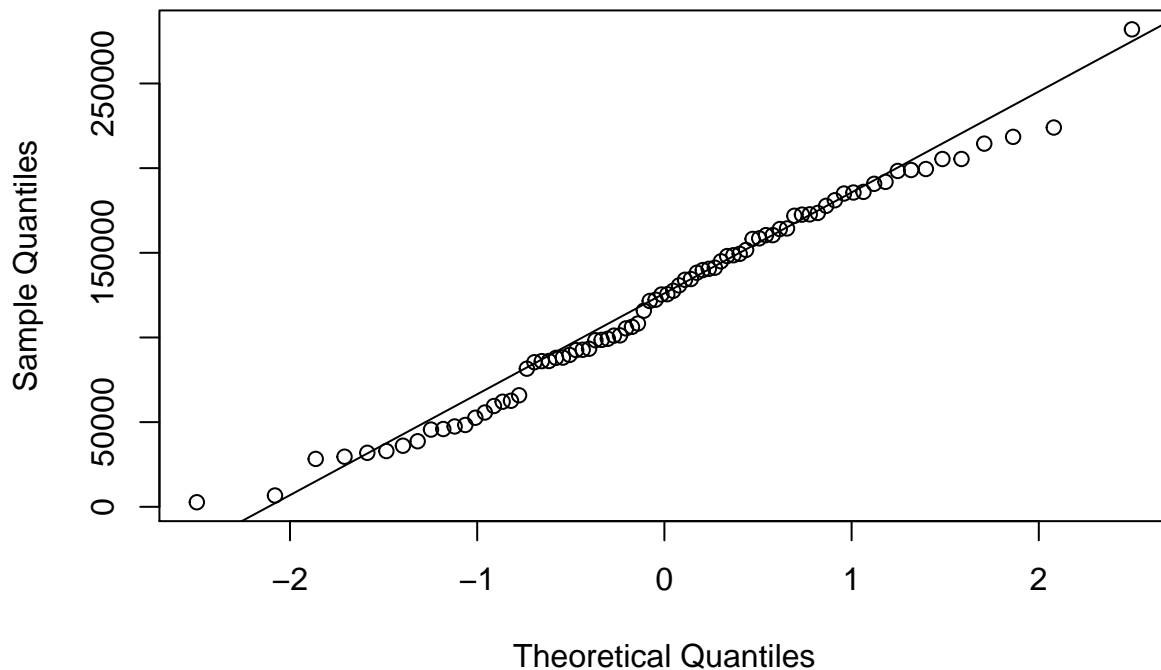
```
ggplot(CD_only) +
  geom_histogram(aes(x = Construction.and.Demolition.Waste)) +
  xlab("Quarterly C&D (U.S. Tons)") +
  ylab("Count") +
  ggtitle("Count of Quarterly C&D Quantities")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
qqnorm(CD_only$Construction.and.Demolition.Waste); qqline(CD_only$Construction.and.Demolition.Waste) #
```

Normal Q-Q Plot



```
# use Pettitt's test (nonparametric) to determine whether there is a shift in the central tendency of t
pettitt.test(CD_only$Construction.and.Demolition.Waste) # change point at time 40
```

```
##
## Pettitt's test for single change-point detection
##
## data: CD_only$Construction.and.Demolition.Waste
## U* = 1396, p-value = 3.2e-10
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                                40
```

```
# Run separate Mann-Kendall for each section
mk.test(CD_only$Construction.and.Demolition.Waste[1:40])
```

```
##
## Mann-Kendall trend test
##
## data: CD_only$Construction.and.Demolition.Waste[1:40]
## z = 1.736, n = 40, p-value = 0.08256
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##          S          varS          tau
## 150.0000000 7366.6666667  0.1923077
```

```
mk.test(CD_only$Construction.and.Demolition.Waste[41:80])
```

```

##
## Mann-Kendall trend test
##
## data: CD_only$Construction.and.Demolition.Waste[41:80]
## z = 2.4817, n = 40, p-value = 0.01308
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##      S      varS      tau
## 214.000000 7366.666667 0.274359

# Is there a second change point?
pettitt.test(CD_only$Construction.and.Demolition.Waste[41:80])

##
## Pettitt's test for single change-point detection
##
## data: CD_only$Construction.and.Demolition.Waste[41:80]
## U* = 203, p-value = 0.04614
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                                27

# position 27, so 41+27 = change point at time 68

# Run separate Mann-Kendall for new section
mk.test(CD_only$Construction.and.Demolition.Waste[69:80]) # p-value = 0.9453, not likely a 3rd change p

##
## Mann-Kendall trend test
##
## data: CD_only$Construction.and.Demolition.Waste[69:80]
## z = 0.068573, n = 12, p-value = 0.9453
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##      S      varS      tau
## 2.00000000 212.66666667 0.03030303

# Is there a third change point?
pettitt.test(CD_only$Construction.and.Demolition.Waste[69:80]) # p-value = p-value = 1.261, no 3rd chan

##
## Pettitt's test for single change-point detection
##
## data: CD_only$Construction.and.Demolition.Waste[69:80]
## U* = 12, p-value = 1.261
## alternative hypothesis: two.sided
## sample estimates:
## probable change point at time K
##                                6

# years corresponding to changepoints
changeoint1 <- CD_only$Quarter.End.Date[40] # between Q4 2007 & Q1 2008 = ~ 2008-02-14
changeoint2 <- CD_only$Quarter.End.Date[68] # between Q4 2014 & Q1 2015 = ~ 2015-02-14

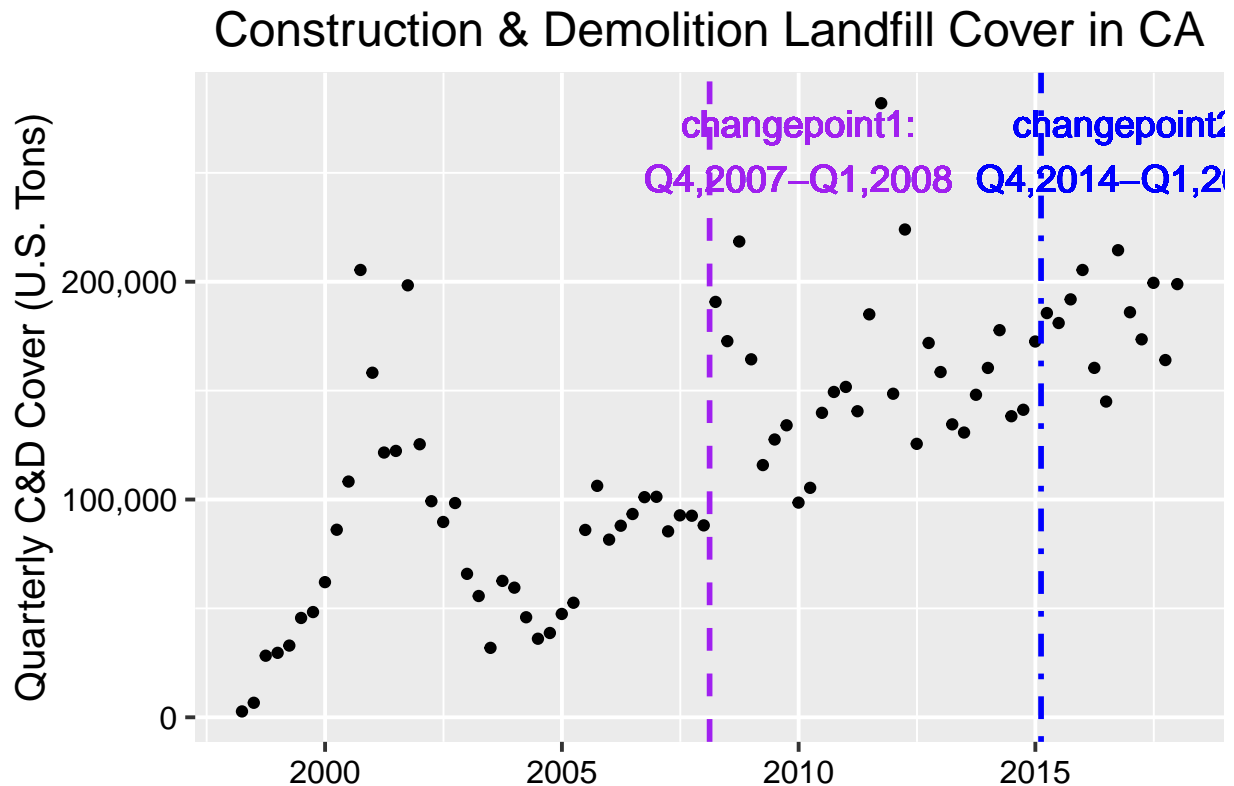
# Add vertical lines to the original graph to represent change points
CD_plot_changeoints <- ggplot(CD_only, aes(x=Quarter.End.Date, y=Construction.and.Demolition.Waste)) +

```

```

geom_point() +
geom_vline(aes(xintercept=as.Date('2008-02-14')), linetype=2, colour="purple", size=1) +
geom_vline(aes(xintercept=as.Date('2015-02-14')), linetype=4, colour="blue", size=1) +
geom_text(x=as.Date('2010-1-1'), y=260000, label=stringr::str_wrap('change point1: Q4,2007-Q1,2008', 10)) +
geom_text(x=as.Date('2017-1-1'), y=260000, label=stringr::str_wrap('change point2: Q4,2014-Q1,2015', 10)) +
xlab('') +
ylab('Quarterly C&D Cover (U.S. Tons)') +
scale_y_continuous(labels = scales::comma) +
ggtitle('Construction & Demolition Landfill Cover in CA')
print(CD_plot_changepoints)

```



```

# save figure
ggsave("CD_plot_changepoints.jpg", CD_plot_changepoints, path = "../Output", height = 4, width = 11, un

```

Misc code for Test 3:

Run separate seasonal Mann-Kendall for each change point

```
CD_as_ts <- ts(CD_only$Quarter.End.Date, start = 1998-03-31, end = 2017-12-31, frequency = 4) # convert
vector of CD quantities into class ts
```

```
smk.test(ts(CD_as_ts[1:39], start = 1998-03-31, end = 2007-09-30, frequency = 4)) # p-value = 3.573e-05
inferring that there is monotonic trend over time with reporting season smk.test(ts(CD_as_ts[40:80], start
= 2007-12-31, end = 2017-12-31, frequency = 4)) SKO: decided not to use bc the fractions were smaller &
smaller as you check change points, making the sample size smaller, which is worse for Mann-Kendall
```

SKO: sample size is 22 for each group. used parametric (instead of non-parametric bc <https://blog.minitab.com/blog/adventures-in-statistics-2/choosing-between-a-nonparametric-test-and-a-parametric-test>)

Format as an aov

```
ADC_quarter_anova <- aov(ADC_total_onlyTotal ADC_total_onlyReport.Quarter) ADC_quarter_anova
summary(ADC_quarter_anova)
```

Run a post-hoc test for pairwise differences

```
TukeyHSD(ADC_quarter_anova) # none of the p values are < 0.05 plot(TukeyHSD(ADC_quarter_anova))
# all of the bars overlap # none of the pairings have significantly different means
```

try Mann-Kendall non-parametric test to detect monotonic trends (H0: there is no trend)

```
total_oldest_to_newest <- ADC_fulldate %>% select(Quarter.End.Date, Total) %>% arrange(Quarter.End.Date)
# arrange data from oldest to newest

mk.test(total_oldest_to_newest$Total) # p-value = 2.326e-09 inferring that there is a monotonic trend over
time
```

run seasonal Mann-Kendall

```
total_as_ts <- ts(total_oldest_to_newest$Total, start = 1995-03-31, end = 2017-12-31, frequency = 4) #
convert total vector into class ts smk.test(total_as_ts) # p-value < 2.2e-16 inferring that there is monotonic
trend over time with reporting season
```

SKO: create figures separately, then grid arrange

Ash

```
ADC_gathered_Ash <- ADC_gathered %>% filter(Type == 'Ash') quarterlyvalues_byyear_plot_Ash <-
ggplot(ADC_gathered_Ash) + geom_boxplot(aes(x = Report.Year, y = Quantity, group = Report.Year))
print(quarterlyvalues_byyear_plot_Ash)
```

group colors by 5 yr chunks: 1998-2002 (magenta), 2003-2007 (turquoise), 2008-2012 (red), 2013-2017 (yellow)

```
quarterlyvalues_alltypes_plot2 <- ggplot(ADC_gathered) + geom_point(data = subset(ADC_gathered,
Report.Year < 2008), aes(x = Report.Quarter, y = Quantity, shape = as.factor(Type), color =
as.factor(Report.Year < 2008), group = Report.Year)) #+ scale_color_manual(values = c('Report.Year <
2008' = 'magenta3'))

print(quarterlyvalues_alltypes_plot2)
```


<https://stackoverflow.com/questions/44915362/custom-grouping-for-legend-in-ggplot>