AFS505 Module 1 Final Exam

William Ottenheimer

2022-09-26

Question 1

What are the basic R data structures? What are the differences between them? In what context would you use one versus the other

The different basic data types in R are Characters, Numeric (both real and decimal), Integers, Logical (TRUE/FALSE), and complex.

Common basic data structures are vectors, Matrices, Lists, and Data Frames.

A vector is a sequence of data elements that must all be the same type. Vectors are technically one dimensional arrays. I would use a vector to store a simple collection of related data. Lists are a kind of vector technically, but most of the time I work with atomic vectors where the data type is restricted. One of the ways I often use vectors is to populate the data for matrices since the mat() function takes a data vector, character string, or list as its arguments. If you attempt to mix different data types in a vector r will coerce the data into the same type.

A matrix is a collection of data elements arranged into a two dimensional rectangular layout with both rows and columns. In this way they are a natural extension of atomic vectors with the additional quality of having dimensionality. Vectors and Matrices require that the data you include be the same data type. A vector is what you should use if your data is one dimensional. You would use a matrix if your data was multidimensional since a matrix has both columns and rows to organize data into.

A data frame is a list of vectors of equal length. it can handle data of different types if you have a mixture of string, logical, and numerical variables. If you have data with different types you should use a list or a data frame instead of a vector/matrix. A data frame is basically a data table. I use data frames to organize and interpret data sets with different variables. The function to fit linear models takes data frames as potential arguments so data frames are good for regression analysis. I often default to data frames for cleaning my data by assigning subsets of old data frames to new data frames however this practice can be cluttered and waste valuable memory. Data frames have attributes like rownames() which is helpful for enhancing the readability of your code. The read family of functions (read.csv or read.table for example) output data frames which is perhaps one of the most important uses of the data frame structure.

A list is the most generic data type. It is basically a vector of objects which can be any type of r object even data structures. A list can even have other lists as its elements. Lists are incredibly useful for working with certain functions in the apply family like sapply and lappy which can take lists or vectors as arguments. The base apply function takes a matrix as an input. Beyond "apply" the way functions in R work is very compatible with lists since functions return single objects a series of function results can be banded together in a list for a function to return.

Question 2

You are provided a folder with three location (county) names, each of which has subfolders for one or two crops, which in turn has a data file.

I think it's easiest to combine steps a and b for question 2 ### Part a and b {.tabset} a iterate through the folders to deal all the files and merge them into a single data frame. You can use a "loop" to iterate or for efficiency check out the list.files() function

b add four additional columns to the merged dataframe corresponding to the county name, crop name, latitude and longitude of the data. You must get this information from the directroy structure you are looping through or the strings returned by the call to list.files()

```
### use list files function to get string vectors of the filenames
### we will use fn as shorthand for filenames

fnlocations <- list.files(path = "C:/Users/otten/Desktop/AFS Repository/Will-O-AFS505/CropModelResults")
print(fnlocations)
```

```
## [1] "Okanogan"   "WallaWalla" "Yakima"
```

```
fncrops <- list.files(paste0("C:\\Users\\otten\\Desktop\\AFS Repository\\Will-O-AFS505\\CropModelResult
print(fncrops)
```

```
## [1] "Corn_grain"   "Winter_Wheat"
```

```
fncords <- list.files(paste0("C:\\Users\\otten\\Desktop\\AFS Repository\\Will-O-AFS505\\CropModelResult
print(fncords)
```

```
## [1] "48.15625N119.71875W" "48.96875N119.65625W"
```

```
### Create an initial data frame to rbind the rest of the data frames to when we read them
### In this step we must also create the new columns appropriately using information from the list.file

Q2maindf <- read.csv(paste0("C:\\Users\\otten\\Desktop\\AFS Repository\\Will-O-AFS505\\CropModelResults

lllist <- strsplit(fncords[1],"N")
latlong <- lllist[[1]]
latlong[1] <- paste0(latlong[1],"N")
latlong
```

```
## [1] "48.15625N"  "119.71875W"
```

```
Q2maindf$countyname <- fnlocations[1]
Q2maindf$cropname <- fncrops[1]
Q2maindf$latitude <- latlong[1]
Q2maindf$longitude <- latlong[2]
### All of this is to make the code general so as to not hard code the known length of the file lists
loclength <- length(fnlocations)
croplength <- length(fncrops)
cordlength <- length(fncords)
### Create a for loop to cycle through each location
### within that loop create a for loop to cycle through each crop
### within that loop create a for loop to cycle through each lat/long

### The loop should read a csv file and store it as a data frame. The loop should then create new colum

for (a in 1:loclength) {
  for(b in 1:croplength){
    for (c in 1:cordlength) {
       fncords <- list.files(paste0("C:\\Users\\otten\\Desktop\\AFS Repository\\Will-O-AFS505\\CropModel
       tempdf <- read.csv(paste0("C:\\Users\\otten\\Desktop\\AFS Repository\\Will-O-AFS505\\CropModelRes
       ### I like to print a piece of my loop just so I can see it is working while I run the code... T
```

```
       print(fncords[c])
       lllist <- strsplit(fncords[c],"N")
       latlong <- lllist[[1]]
       latlong[1] <- paste0(latlong[1],"N")
      tempdf$countyname <- fnlocations[a]
      tempdf$cropname <- fncrops[b]
      tempdf$latitude <- latlong[1]
      tempdf$longitude <- latlong[2]
      Q2maindf <- rbind(Q2maindf,tempdf)
    }
  }

}
```

```
## [1] "48.15625N119.71875W"
## [1] "48.96875N119.65625W"
## [1] "48.15625N119.71875W"
## [1] "48.53125N119.59375W"
## [1] "46.03125N118.21875W"
## [1] "46.28125N118.65625W"
## [1] "46.03125N118.21875W"
## [1] "46.03125N118.40625W"
## [1] "46.15625N119.96875W"
## [1] "46.21875N119.34375W"
## [1] "46.15625N119.34375W"
## [1] "46.21875N119.34375W"
```

```
### remove the duplicate values since one file was read twice
Q2maindf <- Q2maindf[!duplicated(Q2maindf),]
```

Part c

c Rename the column irrig to irrigation_demand and precip to precipitation and export the dataframe as a csv file

```
### Rename the columns
colnames(Q2maindf)[6] = "irrigation_demand"
colnames(Q2maindf)[7] = "precipitation"

### Export the data frame
write.csv(Q2maindf, "C:\\Users\\otten\\Desktop\\AFS Repository\\Will-O-AFS505\\ManagedCropData.csv")
```

Part d

d summarize the annual irrigation demand by crop name and county name

```
### To summarize by other variables I can convert the character data into factor data
Q2maindf$countyname <- as.factor(Q2maindf$countyname)
Q2maindf$cropname <- as.factor(Q2maindf$cropname)

### summarize irrigation demand by county name
tapply(Q2maindf$irrigation_demand,INDEX = Q2maindf$countyname, summary)
```

```
## $Okanogan
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   272.0   363.6   392.0   390.8   416.7   490.5
```

```
##
## $WallaWalla
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   180.0   300.0   360.0   357.0   417.5   520.0
##
## $Yakima
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   390.0   520.0   560.0   545.9   580.0   640.0
```

```
### summarize irrigation demand by crop name
tapply(Q2maindf$irrigation_demand,INDEX = Q2maindf$cropname, summary)
```

```
## $Corn_grain
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   180.0   360.0   413.3   422.4   480.0   640.0
##
## $Winter_Wheat
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   208.0   363.9   420.1   440.3   540.0   640.0
```

Part e

e What is the average yield of Winter Wheat in Walla Walla at 46.03125N118.40625W for the year ranges (1981-1990), (1991-2000), and (2001-2019)?

The instructions don't specify how I need to calculate this information so I am using excel to filter the data quickly and report the average. This is easy because earlier in the exam I exported the data frame as a csv file

For the crop at that location during the years 1981-1990 the average yield is 7660.89502

For the Crop at that location during the years 1991-2000 the average yield is 8086.688721

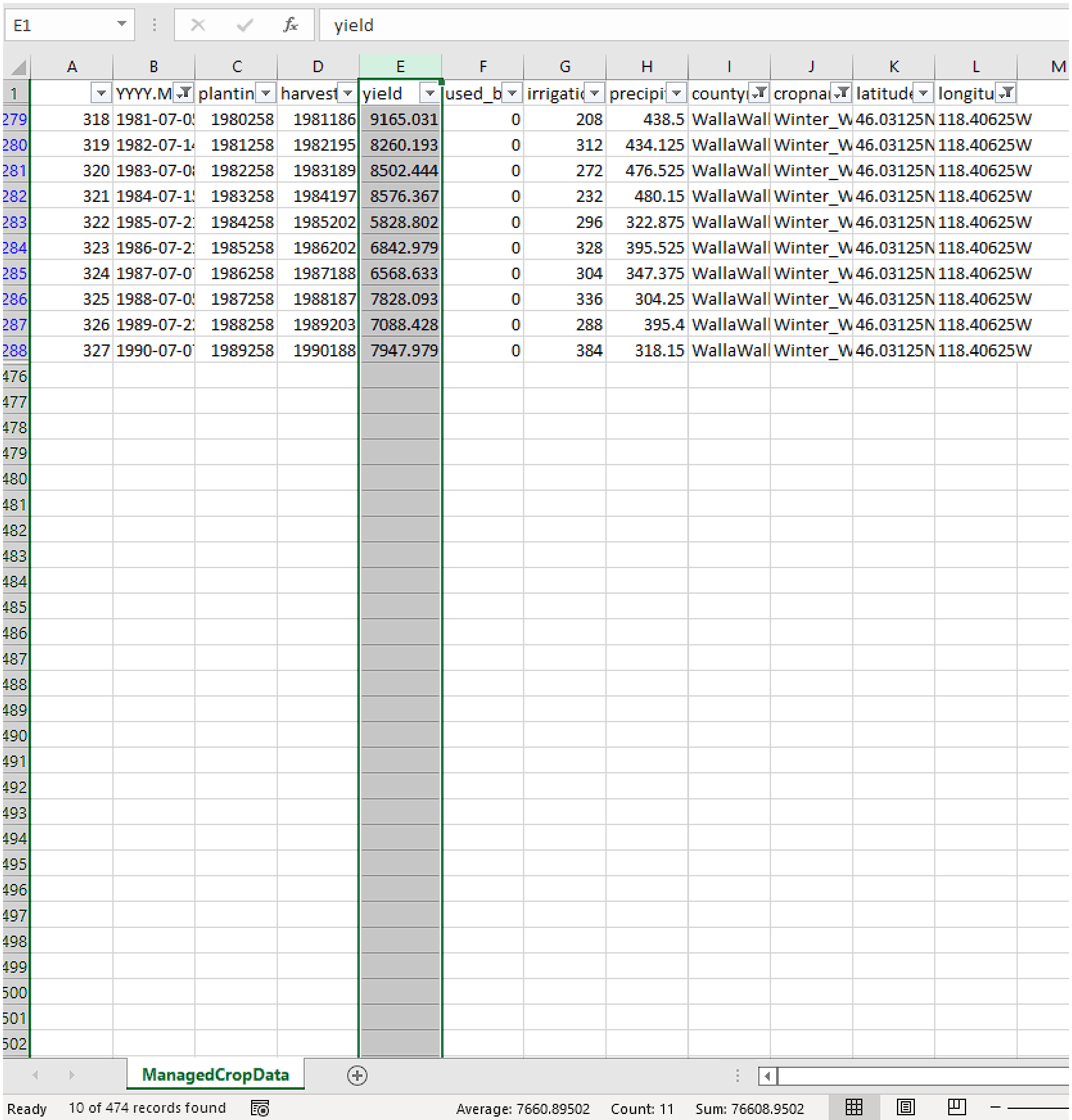
For the Crop at that location during the years 2001-2019 the average yield is 7720.68303

Part f

f which location has the highest yield (average) for the time period (2001-2019) for grain corn

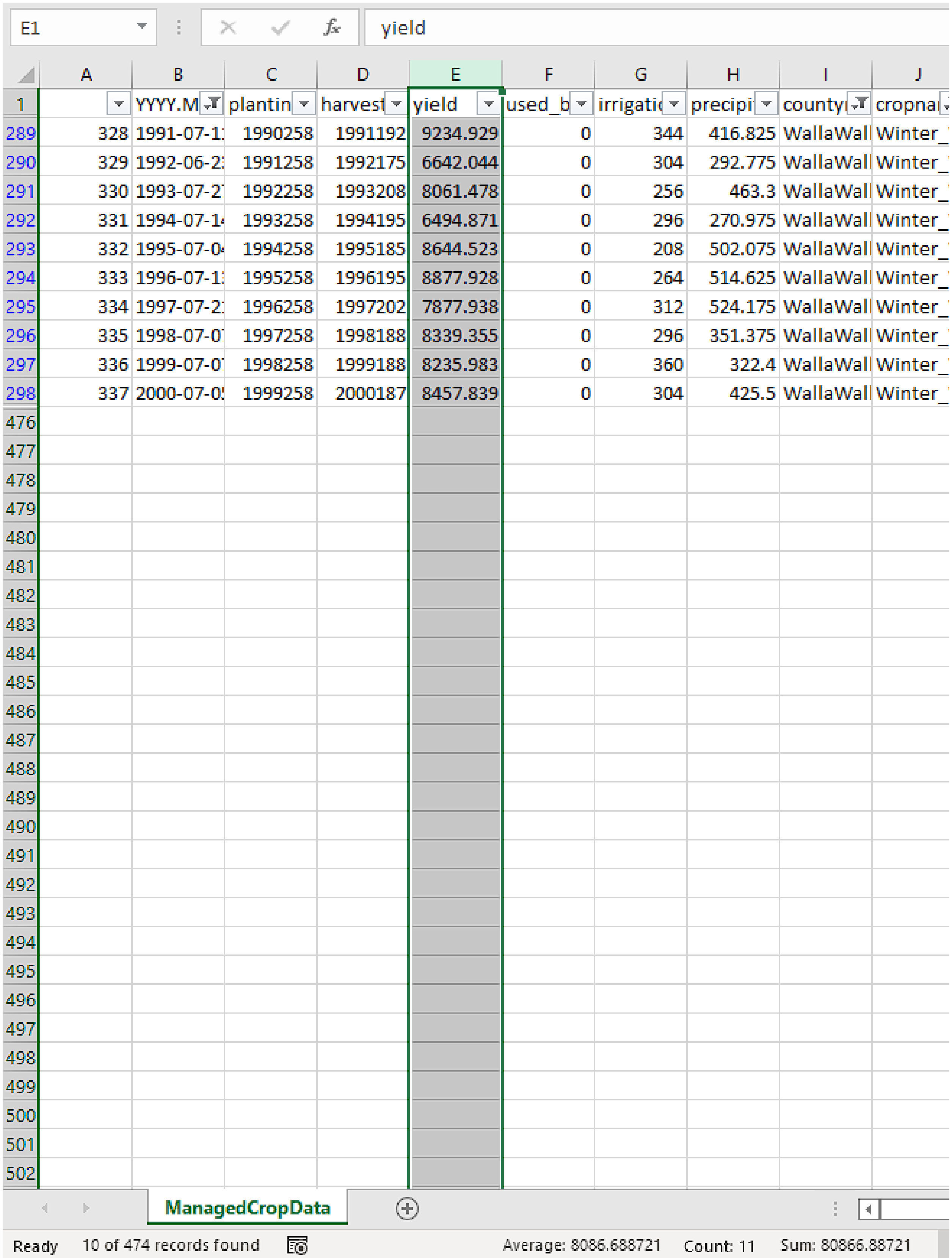
Again because the instructions don't specify I must do this in r I will make the calculations in excel

```
### I will create a subset of my data to figure this out first I am only interested in values related t
library(stringr)

Q2fsub <- subset(Q2maindf ,Q2maindf$cropname == "Corn_grain")

### Secondly I am only interested in the years 2001-2019. It will be easier for me to subset this data

Q2fsub$YYYY.MM.DD.DOY. <- str_trunc(Q2fsub$YYYY.MM.DD.DOY., 4,"right", ellipsis = "")
Q2fsub$YYYY.MM.DD.DOY. <- as.numeric(Q2fsub$YYYY.MM.DD.DOY.)
Q2fsub <- subset(Q2fsub, Q2fsub$YYYY.MM.DD.DOY. >= 2001)

### I will coerce the location (lat long) into factors so that I can calculate the means for each area

Q2fsub$latitude <- as.factor(Q2fsub$latitude)
Q2fsub$longitude <- as.factor(Q2fsub$longitude)

### I use the tapply function to compute means
tapply(Q2fsub$yield, INDEX = Q2fsub$latitude, mean)
```



	A	B	C	D	E	F	G	H	I	J	K	L
1		YYYY.M	plantin	harvest	yield	used_b	irrigati	precipi	county	cropna	latitude	longitu
279	318	1981-07-0	1980258	1981186	9165.031	0	208	438.5	WallaWall	Winter_W	46.03125N	118.40625W
280	319	1982-07-1	1981258	1982195	8260.193	0	312	434.125	WallaWall	Winter_W	46.03125N	118.40625W
281	320	1983-07-0	1982258	1983189	8502.444	0	272	476.525	WallaWall	Winter_W	46.03125N	118.40625W
282	321	1984-07-1	1983258	1984197	8576.367	0	232	480.15	WallaWall	Winter_W	46.03125N	118.40625W
283	322	1985-07-2	1984258	1985202	5828.802	0	296	322.875	WallaWall	Winter_W	46.03125N	118.40625W
284	323	1986-07-2	1985258	1986202	6842.979	0	328	395.525	WallaWall	Winter_W	46.03125N	118.40625W
285	324	1987-07-0	1986258	1987188	6568.633	0	304	347.375	WallaWall	Winter_W	46.03125N	118.40625W
286	325	1988-07-0	1987258	1988187	7828.093	0	336	304.25	WallaWall	Winter_W	46.03125N	118.40625W
287	326	1989-07-2	1988258	1989203	7088.428	0	288	395.4	WallaWall	Winter_W	46.03125N	118.40625W
288	327	1990-07-0	1989258	1990188	7947.979	0	384	318.15	WallaWall	Winter_W	46.03125N	118.40625W

Figure 1: Excel computation of first average



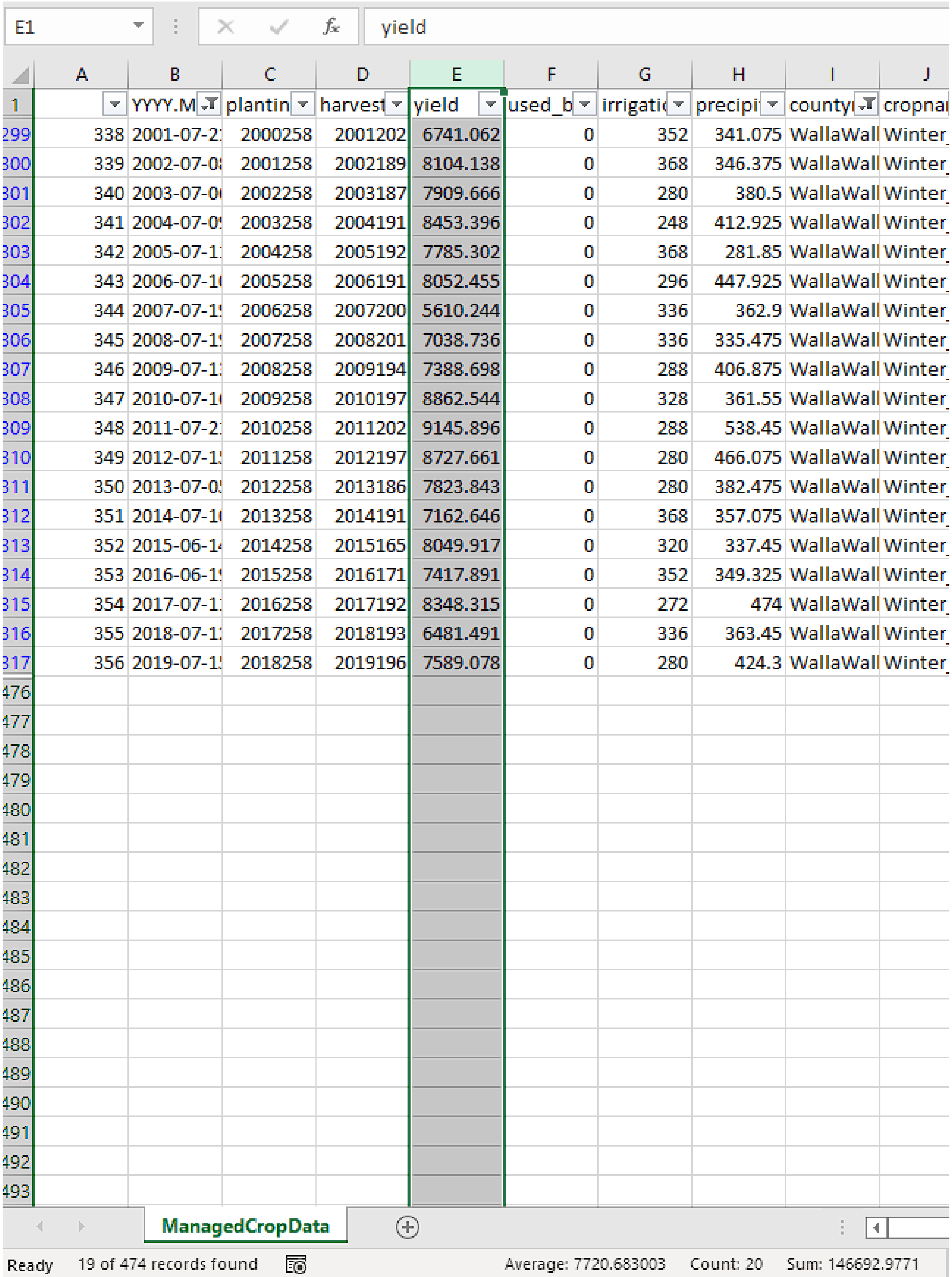
E1 | yield

	A	B	C	D	E	F	G	H	I	J
1		YYYY.M	plantin	harvest	yield	used_b	irrigati	precipi	county	cropna
289	328	1991-07-1	1990258	1991192	9234.929	0	344	416.825	WallaWall	Winter_
290	329	1992-06-2	1991258	1992175	6642.044	0	304	292.775	WallaWall	Winter_
291	330	1993-07-2	1992258	1993208	8061.478	0	256	463.3	WallaWall	Winter_
292	331	1994-07-1	1993258	1994195	6494.871	0	296	270.975	WallaWall	Winter_
293	332	1995-07-0	1994258	1995185	8644.523	0	208	502.075	WallaWall	Winter_
294	333	1996-07-1	1995258	1996195	8877.928	0	264	514.625	WallaWall	Winter_
295	334	1997-07-2	1996258	1997202	7877.938	0	312	524.175	WallaWall	Winter_
296	335	1998-07-0	1997258	1998188	8339.355	0	296	351.375	WallaWall	Winter_
297	336	1999-07-0	1998258	1999188	8235.983	0	360	322.4	WallaWall	Winter_
298	337	2000-07-0	1999258	2000187	8457.839	0	304	425.5	WallaWall	Winter_

ManagedCropData

Ready 10 of 474 records found Average: 8086.688721 Count: 11 Sum: 80866.88721

Figure 2: Excel computation of second average



	A	B	C	D	E	F	G	H	I	J
1		YYYY.M	plantin	harvest	yield	used_b	irrigatio	precipi	county	cropna
299	338	2001-07-2	2000258	2001202	6741.062	0	352	341.075	WallaWall	Winter_
300	339	2002-07-0	2001258	2002189	8104.138	0	368	346.375	WallaWall	Winter_
301	340	2003-07-0	2002258	2003187	7909.666	0	280	380.5	WallaWall	Winter_
302	341	2004-07-0	2003258	2004191	8453.396	0	248	412.925	WallaWall	Winter_
303	342	2005-07-1	2004258	2005192	7785.302	0	368	281.85	WallaWall	Winter_
304	343	2006-07-1	2005258	2006191	8052.455	0	296	447.925	WallaWall	Winter_
305	344	2007-07-1	2006258	2007200	5610.244	0	336	362.9	WallaWall	Winter_
306	345	2008-07-1	2007258	2008201	7038.736	0	336	335.475	WallaWall	Winter_
307	346	2009-07-1	2008258	2009194	7388.698	0	288	406.875	WallaWall	Winter_
308	347	2010-07-1	2009258	2010197	8862.544	0	328	361.55	WallaWall	Winter_
309	348	2011-07-2	2010258	2011202	9145.896	0	288	538.45	WallaWall	Winter_
310	349	2012-07-1	2011258	2012197	8727.661	0	280	466.075	WallaWall	Winter_
311	350	2013-07-0	2012258	2013186	7823.843	0	280	382.475	WallaWall	Winter_
312	351	2014-07-1	2013258	2014191	7162.646	0	368	357.075	WallaWall	Winter_
313	352	2015-06-1	2014258	2015165	8049.917	0	320	337.45	WallaWall	Winter_
314	353	2016-06-1	2015258	2016171	7417.891	0	352	349.325	WallaWall	Winter_
315	354	2017-07-1	2016258	2017192	8348.315	0	272	474	WallaWall	Winter_
316	355	2018-07-1	2017258	2018193	6481.491	0	336	363.45	WallaWall	Winter_
317	356	2019-07-1	2018258	2019196	7589.078	0	280	424.3	WallaWall	Winter_

Figure 3: Excel computation of third average

```
## 46.03125N 46.15625N 46.21875N 46.28125N 48.15625N 48.96875N
##  9981.876 13536.885 15263.030 12482.990 14092.233 14139.643
```

```
tapply(Q2fsub$yield, INDEX = Q2fsub$longitude, mean)
```

```
## 118.21875W 118.65625W 119.34375W 119.65625W 119.71875W 119.96875W
##   9981.876  12482.990  15263.030  14139.643  14092.233  13536.885
```

It is apparent that 46.21875N119.34375W has the highest average yield for grain corn in the period 2001-20019 with an average of 15263.030

Github link

Repository link: https://github.com/Will-Ottenheimer/Will-O-AFS505.git

Rmarkdown Permalink: https://github.com/Will-Ottenheimer/Will-O-AFS505/blob/4f6e32785add3316e0c4f013e6d832c7e1244ae9/AFS505%20Final%20Exam.Rmd

CSV File Permalink: https://github.com/Will-Ottenheimer/Will-O-AFS505/blob/4f6e32785add3316e0c4f013e6d832c7e1244ae9/ManagedCropData.csv

Question 3

Was the data provided to you well described? If not, what information was missing? Comment on what kind of meta data (description about the data) should be included as a best practice while sharing data sets.

The data was poorly described. To begin the unit of measurement is missing for nearly all the variables. I don't know if yield is in bushels or grams, or truckloads so the meta data file should explain the measurement unit for all the numeric data. The Used_biomass is not labeled as a dummy variable (if it is it should be). The meta data should express what format the planting and harvesting date are in.

The best meta data includes the date the data was created and the date it was last modified. Metadata should also include the name of the author of the data set and their contact info. Additionally, quality metadata should have tags and categories and titles and descriptions for all the data. Databases in particular should include information about the data types, columns, constraints, and relationships between the tabular data.

Question 4

Create an R Markdown file with different tabs for each of the six parts of question 2. In a seventh tab add the github link which has your Rscript/R markdownfile and the csv file generated from 2(c)

Because I did steps a and b of question 2 in one step there will only be 6 tabs.