R Notebook

Lab authored by Joanna Lankester, PhD for CS 48A

Lab completed by April Salazar

Introduction

In this lab, we'll use tidyverse - more specifically, several dplyr library functions - to extract data in the format we need for our visualizations. This data transformation will allow us to create plots that we can't plot from the raw data, for example, from subsets of data or from data that has been grouped by a particular trait and aggregated.

This lab will require saving some results to variables where specified. Let's try it; run the following code to save the value 5 to the variable Q0:

```
Q0 <- 5
```

Now run this to check the value of Q0:

```
Q0
```

```
## [1] 5
```

Note that we could also use the equals sign to save the value 5 to Q0. Additionally, we could use one code chunk to both assign the variable and check the value by putting the Q0 check at the end of the chunk. Run this to check that both of these claims are true:

```
Q0 = 5
Q0
```

```
## [1] 5
```

Instructions

- Run each chunk of code in order.
- Some steps will require code and will be marked with an asterisk*. Others will require a written response and will be marked with an asterisk and (response), like this: *(response)
- Important: where specified, please be sure to assign the variable exactly as named. For example, the above chunks required assigning to a variable Q0, which was capital Q followed by 0. Please do not name this as q0, Q_0, Q00, or anything else that isn't literally Q0.
- Where not specified, you do not need to assign the result to a variable.

Q1. Run the following chunk to load the libraries needed for this lab.

There may be some warnings and messages about conflicts. These are ok as long as the library loads.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
```

```
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
# the following is being defined so that you can use it later with plots, if you wish
larger_font <- theme(axis.title = element_text(size=14)) +
    theme(axis.text = element_text(size=12))
```

Q2. Run the following chunk to load the datasets needed for this lab.

```
df_tips = tips %>%
    mutate(tip_percent = tip/total_bill*100)

states <- cbind(state.abb,state.region,state.division,data.frame(state.x77)) %>%
    rename(region=state.region,division=state.division,abb=state.abb,income_per_capita = Income) %>%
    mutate(income = income_per_capita * Population)
```

Part 1

- Main goal: use filtering to allow plotting of a subset of data
- Secondary goal: practice creating distribution plots

Q3. Run this to see what the df_tips data frame looks like.

```
df_tips %>%
    head()
```

```
##   total_bill  tip    sex smoker day   time size tip_percent
## 1      16.99 1.01 Female     No Sun Dinner    2    5.944673
## 2      10.34 1.66   Male     No Sun Dinner    3   16.054159
## 3      21.01 3.50   Male     No Sun Dinner    3   16.658734
## 4      23.68 3.31   Male     No Sun Dinner    2   13.978041
## 5      24.59 3.61 Female     No Sun Dinner    4   14.680765
## 6      25.29 4.71   Male     No Sun Dinner    4   18.623962
```

Q4. Run this to see how many rows are in the data frame.

```
df_tips %>%
    tally()
```

```
##     n
```

```
## 1 244
```

Q5*. Get only the restaurant bills in df_tips greater than $20. ***Save the result as Q5.***

```
Q5 <- df_tips %>%
    filter(total_bill > 20)
# you can leave this here to check your work
Q5
```

```
##     total_bill   tip    sex smoker  day   time size tip_percent
## 3        21.01  3.50   Male     No  Sun Dinner    3   16.658734
## 4        23.68  3.31   Male     No  Sun Dinner    2   13.978041
## 5        24.59  3.61 Female     No  Sun Dinner    4   14.680765
## 6        25.29  4.71   Male     No  Sun Dinner    4   18.623962
## 8        26.88  3.12   Male     No  Sun Dinner    4   11.607143
## 12       35.26  5.00 Female     No  Sun Dinner    4   14.180374
## 16       21.58  3.92   Male     No  Sun Dinner    2   18.164968
## 20       20.65  3.35   Male     No  Sat Dinner    3   16.222760
## 22       20.29  2.75 Female     No  Sat Dinner    2   13.553475
## 24       39.42  7.58   Male     No  Sat Dinner    4   19.228818
## 29       21.70  4.30   Male     No  Sat Dinner    2   19.815668
## 34       20.69  2.45 Female     No  Sat Dinner    4   11.841469
## 36       24.06  3.60   Male     No  Sat Dinner    3   14.962594
## 40       31.27  5.00   Male     No  Sat Dinner    3   15.989767
## 45       30.40  5.60   Male     No  Sun Dinner    4   18.421053
## 47       22.23  5.00   Male     No  Sun Dinner    2   22.492128
## 48       32.40  6.00   Male     No  Sun Dinner    4   18.518519
## 49       28.55  2.05   Male     No  Sun Dinner    3    7.180385
## 53       34.81  5.20 Female     No  Sun Dinner    4   14.938236
## 55       25.56  4.34   Male     No  Sun Dinner    4   16.979656
## 57       38.01  3.00   Male    Yes  Sat Dinner    4    7.892660
## 58       26.41  1.50 Female     No  Sat Dinner    2    5.679667
## 60       48.27  6.73   Male     No  Sat Dinner    4   13.942407
## 61       20.29  3.21   Male    Yes  Sat Dinner    2   15.820601
## 66       20.08  3.15   Male     No  Sat Dinner    3   15.687251
## 69       20.23  2.01   Male     No  Sat Dinner    2    9.935739
## 73       26.86  3.14 Female    Yes  Sat Dinner    2   11.690246
## 74       25.28  5.00 Female    Yes  Sat Dinner    2   19.778481
## 78       27.20  4.00   Male     No Thur  Lunch    4   14.705882
## 79       22.76  3.00   Male     No Thur  Lunch    2   13.181019
## 84       32.68  5.00   Male    Yes Thur  Lunch    2   15.299878
## 86       34.83  5.17 Female     No Thur  Lunch    4   14.843526
## 89       24.71  5.85   Male     No Thur  Lunch    2   23.674626
## 90       21.16  3.00   Male     No Thur  Lunch    2   14.177694
## 91       28.97  3.00   Male    Yes  Fri Dinner    2   10.355540
## 92       22.49  3.50   Male     No  Fri Dinner    2   15.562472
## 95       22.75  3.25 Female     No  Fri Dinner    2   14.285714
## 96       40.17  4.73   Male    Yes  Fri Dinner    4   11.774956
## 97       27.28  4.00   Male    Yes  Fri Dinner    2   14.662757
## 99       21.01  3.00   Male    Yes  Fri Dinner    2   14.278915
## 103      44.30  2.50 Female    Yes  Sat Dinner    3    5.643341
## 104      22.42  3.48 Female    Yes  Sat Dinner    2   15.521855
## 105      20.92  4.08 Female     No  Sat Dinner    2   19.502868
## 107      20.49  4.06   Male    Yes  Sat Dinner    2   19.814544
```

```
## 108      25.21  4.29   Male    Yes  Sat Dinner    2   17.017057
## 113      38.07  4.00   Male     No  Sun Dinner    3   10.506961
## 114      23.95  2.55   Male     No  Sun Dinner    2   10.647182
## 115      25.71  4.00 Female     No  Sun Dinner    3   15.558149
## 117      29.93  5.07   Male     No  Sun Dinner    4   16.939526
## 120      24.08  2.92 Female     No Thur  Lunch    4   12.126246
## 126      29.80  4.20 Female     No Thur  Lunch    6   14.093960
## 130      22.82  2.18   Male     No Thur  Lunch    3    9.553024
## 132      20.27  2.83 Female     No Thur  Lunch    2   13.961519
## 142      34.30  6.70   Male     No Thur  Lunch    6   19.533528
## 143      41.19  5.00   Male     No Thur  Lunch    5   12.138869
## 144      27.05  5.00 Female     No Thur  Lunch    6   18.484288
## 154      24.55  2.00   Male     No  Sun Dinner    4    8.146640
## 156      29.85  5.14 Female     No  Sun Dinner    5   17.219430
## 157      48.17  5.00   Male     No  Sun Dinner    6   10.379905
## 158      25.00  3.75 Female     No  Sun Dinner    4   15.000000
## 161      21.50  3.50   Male     No  Sun Dinner    4   16.279070
## 166      24.52  3.48   Male     No  Sun Dinner    3   14.192496
## 167      20.76  2.24   Male     No  Sun Dinner    2   10.789981
## 168      31.71  4.50   Male     No  Sun Dinner    4   14.191107
## 171      50.81 10.00   Male    Yes  Sat Dinner    3   19.681165
## 174      31.85  3.18   Male    Yes  Sun Dinner    2    9.984301
## 176      32.90  3.11   Male    Yes  Sun Dinner    2    9.452888
## 180      34.63  3.55   Male    Yes  Sun Dinner    2   10.251227
## 181      34.65  3.68   Male    Yes  Sun Dinner    4   10.620491
## 182      23.33  5.65   Male    Yes  Sun Dinner    2   24.217745
## 183      45.35  3.50   Male    Yes  Sun Dinner    3    7.717751
## 184      23.17  6.50   Male    Yes  Sun Dinner    4   28.053517
## 185      40.55  3.00   Male    Yes  Sun Dinner    2    7.398274
## 186      20.69  5.00   Male     No  Sun Dinner    5   24.166264
## 187      20.90  3.50 Female    Yes  Sun Dinner    3   16.746411
## 188      30.46  2.00   Male    Yes  Sun Dinner    5    6.565988
## 190      23.10  4.00   Male    Yes  Sun Dinner    3   17.316017
## 193      28.44  2.56   Male    Yes Thur  Lunch    2    9.001406
## 198      43.11  5.00 Female    Yes Thur  Lunch    4   11.598237
## 205      20.53  4.00   Male    Yes Thur  Lunch    4   19.483682
## 207      26.59  3.41   Male    Yes  Sat Dinner    3   12.824370
## 208      38.73  3.00   Male    Yes  Sat Dinner    4    7.745933
## 209      24.27  2.03   Male    Yes  Sat Dinner    2    8.364236
## 211      30.06  2.00   Male    Yes  Sat Dinner    3    6.653360
## 212      25.89  5.16   Male    Yes  Sat Dinner    4   19.930475
## 213      48.33  9.00   Male     No  Sat Dinner    4   18.621974
## 215      28.17  6.50 Female    Yes  Sat Dinner    3   23.074192
## 217      28.15  3.00   Male    Yes  Sat Dinner    5   10.657194
## 220      30.14  3.09 Female    Yes  Sat Dinner    4   10.252157
## 228      20.45  3.00   Male     No  Sat Dinner    4   14.669927
## 230      22.12  2.88 Female    Yes  Sat Dinner    2   13.019892
## 231      24.01  2.00   Male    Yes  Sat Dinner    4    8.329863
## 238      32.83  1.17   Male    Yes  Sat Dinner    2    3.563814
## 239      35.83  4.67 Female     No  Sat Dinner    3   13.033771
## 240      29.03  5.92   Male     No  Sat Dinner    3   20.392697
## 241      27.18  2.00 Female    Yes  Sat Dinner    2    7.358352
## 242      22.67  2.00   Male    Yes  Sat Dinner    2    8.822232
```

Q6*. Find the number of restaurant bills in df_tips greater than $20. Save the result as Q6.

Hint: borrow syntax from prior questions.

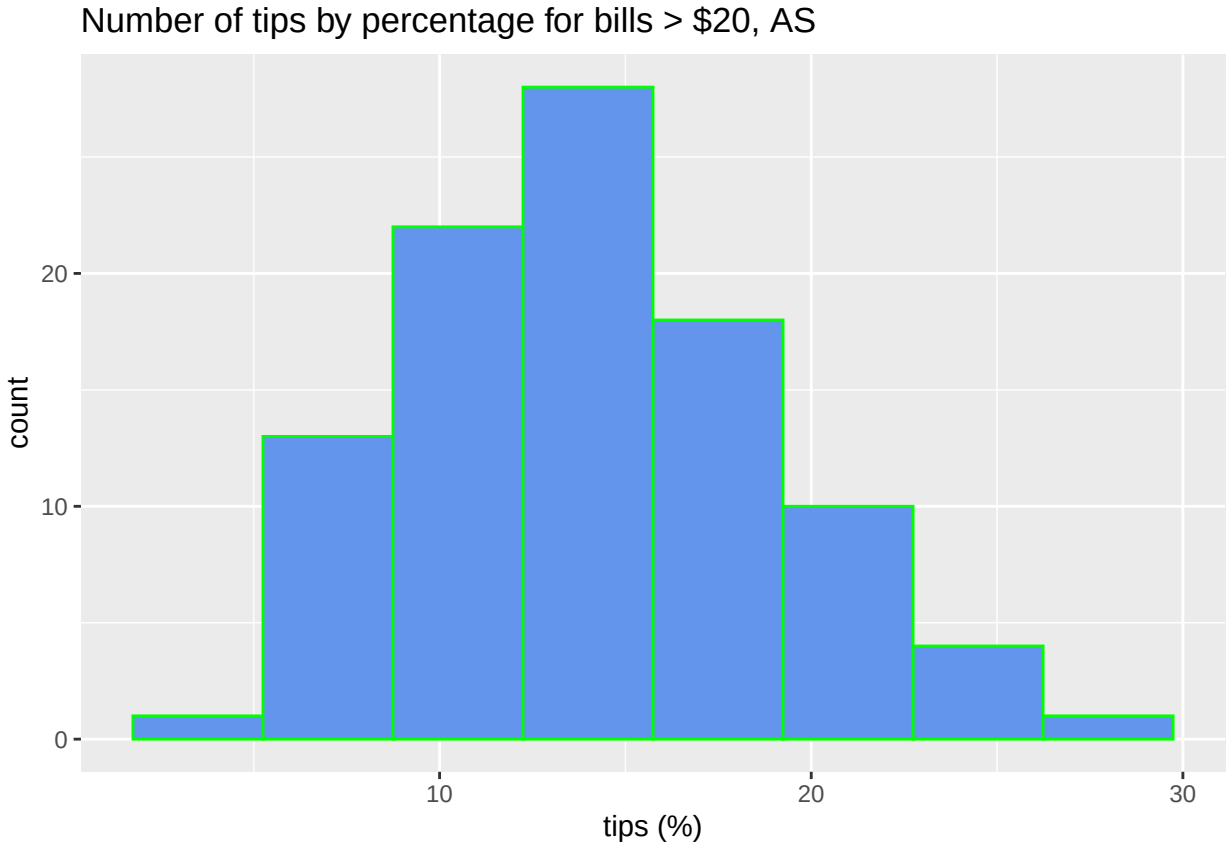
```
Q6 = 97
Q6
```

```
## [1] 97
```

Q7*. Plot a histogram of the tip percent only for restaurant bills greater than $20.

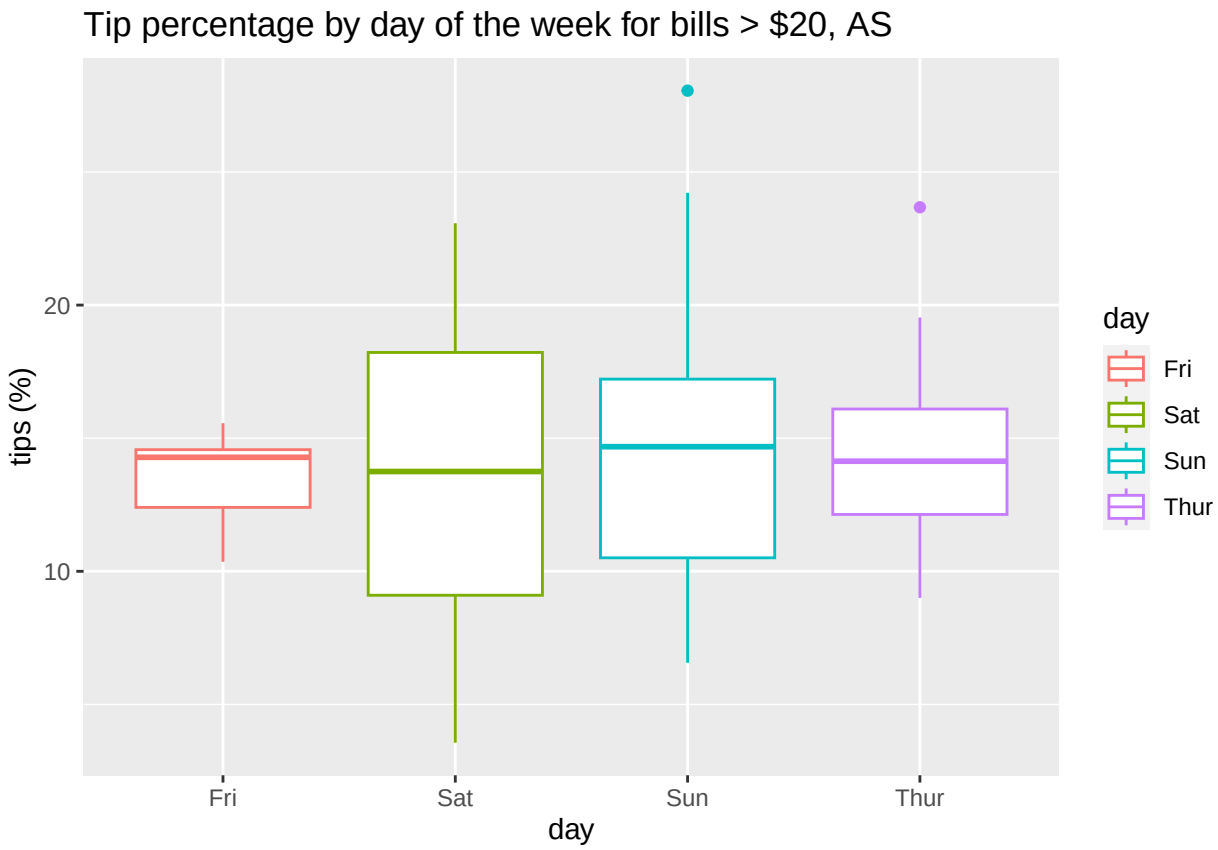
Hint: borrow some of the syntax from a prior question. Remember to add necessary titles and labels.

```
ggplot(Q5,aes(x=tip_percent)) +
    geom_histogram(bins =8, fill = "cornflowerblue", color = "green") +
    labs(title="Number of tips by percentage for bills > $20, AS", x="tips (%)")
```



Q8*. Plot a boxplot of the tip percent by day of the week only for restaurant bills greater than $20. Hint: borrow some of the syntax from a prior question. Remember to add necessary titles and labels.

```
ggplot(Q5,aes(x=day,y=tip_percent,color=day)) +
    geom_boxplot() +
    labs(title="Tip percentage by day of the week for bills > $20, AS", y="tips (%)")
```



Q9*(response only). Interpret the plot from Q8 given the following questions. On which day of the week were the highest tip percentages received, and on which day of the week were the most consistent tip percentages received?

- The highest tips were on **Sunday**.
- The most consistent tips were on **Friday**.

Part 2

- Main goal: Use group_by and aggregation functions to allow more complex tables and plots than we could have made from raw data
- Secondary goal: Get introduced to some additional syntax (top_n, fct_reorder)

Q10. Run this to see what the states data frame looks like.

This data frame contains statistics for different states in 1974.

```
states %>% head()
```

```
##            abb region           division Population income_per_capita
## Alabama     AL  South East South Central       3615              3624
## Alaska      AK   West            Pacific        365              6315
## Arizona     AZ   West           Mountain       2212              4530
## Arkansas    AR  South West South Central       2110              3378
## California  CA   West            Pacific      21198              5114
## Colorado    CO   West           Mountain       2541              4884
##            Illiteracy Life.Exp Murder HS.Grad Frost   Area    income
## Alabama           2.1    69.05   15.1    41.3    20  50708  13100760
## Alaska            1.5    69.31   11.3    66.7   152 566432   2304975
```

```
## Arizona           1.8    70.55    7.8    58.1    15 113417  10020360
## Arkansas          1.9    70.66   10.1    39.9    65  51945   7127580
## California        1.1    71.71   10.3    62.6    20 156361 108406572
## Colorado          0.7    72.06    6.8    63.9   166 103766  12410244
```

Q11*. Make a table of region and sum of income for that region. Call this sum of income column "total_income". Save the table to a variable called Q11.

Hints:

- We need a result *for each* region.
- *For each* generally means we will use *group_by*.

```
Q11 = states %>%
    group_by(region) %>%
    summarize(total_income = sum(income))
Q11
```

```
## # A tibble: 4 x 2
##   region        total_income
##   <fct>                <dbl>
## 1 Northeast        237491085
## 2 South            277449070
## 3 North Central    269154800
## 4 West             185708796
```

Q12*. Get the full row for the top 4 states in population for each region. Save this result as Q12.

To do this, use the top_n function, which will use a syntax like: *top_n(number, variable)* The number will be the top number of rows you want, and the variable will be the variable that should be used for selecting that top number.

```
Q12 = states %>%
    group_by(region) %>%
    top_n(4, Population)
Q12
```

```
## # A tibble: 16 x 12
## # Groups:   region [4]
##    abb   region division Population income_per_capita Illiteracy Life.Exp Murder
##    <chr> <fct>  <fct>         <dbl>             <dbl>      <dbl>    <dbl>  <dbl>
##  1 CA    West   Pacific       21198              5114        1.1     71.7   10.3
##  2 CO    West   Mountain       2541              4884        0.7     72.1    6.8
##  3 FL    South  South A~       8277              4815        1.3     70.7   10.7
##  4 IL    North~ East No~      11197              5107        0.9     70.1   10.3
##  5 IN    North~ East No~       5313              4458        0.7     70.9    7.1
##  6 MA    North~ New Eng~       5814              4755        1.1     71.8    3.3
##  7 MI    North~ East No~       9111              4751        0.9     70.6   11.1
##  8 NJ    North~ Middle ~       7333              5237        1.1     70.9    5.2
##  9 NY    North~ Middle ~      18076              4903        1.4     70.6   10.9
## 10 NC    South  South A~       5441              3875        1.8     69.2   11.1
## 11 OH    North~ East No~      10735              4561        0.8     70.8    7.4
## 12 OR    West   Pacific        2284              4660        0.6     72.1    4.2
## 13 PA    North~ Middle ~      11860              4449        1       70.4    6.1
## 14 TX    South  West So~      12237              4188        2.2     70.9   12.2
## 15 VA    South  South A~       4981              4701        1.4     70.1    9.5
```

```
## 16 WA    West   Pacific        3559              4864        0.6     71.7    4.3
## # i 4 more variables: HS.Grad <dbl>, Frost <dbl>, Area <dbl>, income <dbl>
```

As you've noticed, there are 4 regions in this data frame which have differing amounts of total income. We would like to visualize how much of each region's income comes from the 4 most populous states. We will do this in several steps.

Q13*. Create a table with one column as the region and another column as the sum of the region's income that comes from the top 4 most populous states. Name the second column *top_4_income*. Save the table as a variable Q13.

Hint: this can be done entirely using a combination of syntax from the prior steps.

```
Q13 = states %>%
    group_by(region) %>%
    top_n(4, Population) %>%
    summarize(top_4_income = sum(income))
Q13
```

```
## # A tibble: 4 x 2
##   region        top_4_income
##   <fct>                <dbl>
## 1 Northeast        207440259
## 2 South            135601867
## 3 North Central    173117129
## 4 West             148771232
```

Q14. Run this provided code to merge the two relevant data frames.

```
Q14 <- merge(Q11,Q13,on="region")
Q14
```

```
##          region total_income top_4_income
## 1 North Central    269154800    173117129
## 2     Northeast    237491085    207440259
## 3         South    277449070    135601867
## 4          West    185708796    148771232
```

Q15. Run this provided code to convert the data from its current wide format to long format.

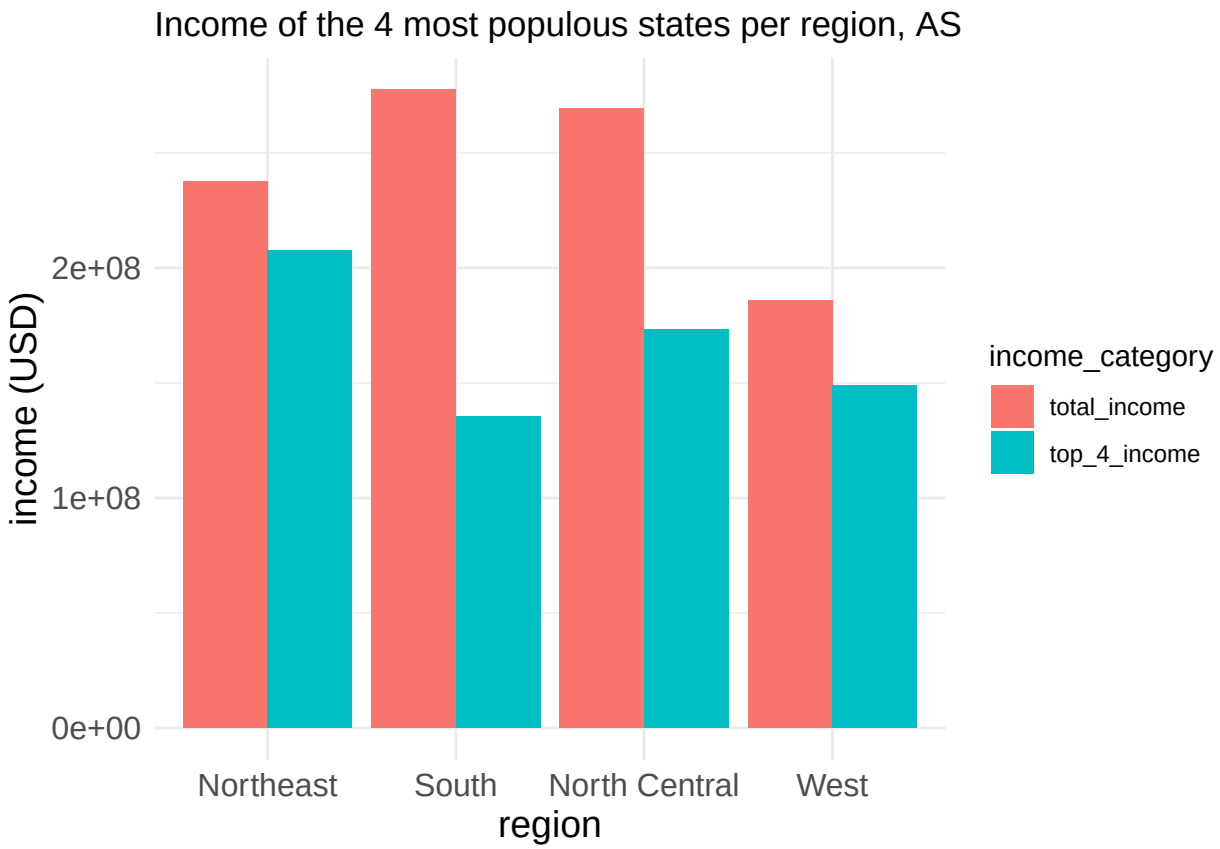
```
Q15 <- melt(Q14,variable.name = "income_category",value.name = "income",id.vars = "region")
Q15
```

```
##          region income_category    income
## 1 North Central    total_income 269154800
## 2     Northeast    total_income 237491085
## 3         South    total_income 277449070
## 4          West    total_income 185708796
## 5 North Central    top_4_income 173117129
## 6     Northeast    top_4_income 207440259
## 7         South    top_4_income 135601867
## 8          West    top_4_income 148771232
```

Q16*. Use the Q15 answer to create a grouped bar plot by region according to the following instructions.

- Each region should have two bars in different colors - one color for total_income, and one color for top_4_income.
- Customize the plot by adding title/labels, adding a theme, and enlarging the font (see Q1; you can reuse the larger_font variable, if you wish).
- We will further improve the plot in the next step.

```
ggplot(Q15,aes(x=region,y=income,fill=income_category)) +
    geom_col(position="dodge") +
    labs(title="Income of the 4 most populous states per region, AS",x="region",y="income (USD)") +
    theme_minimal() +
    larger_font
```

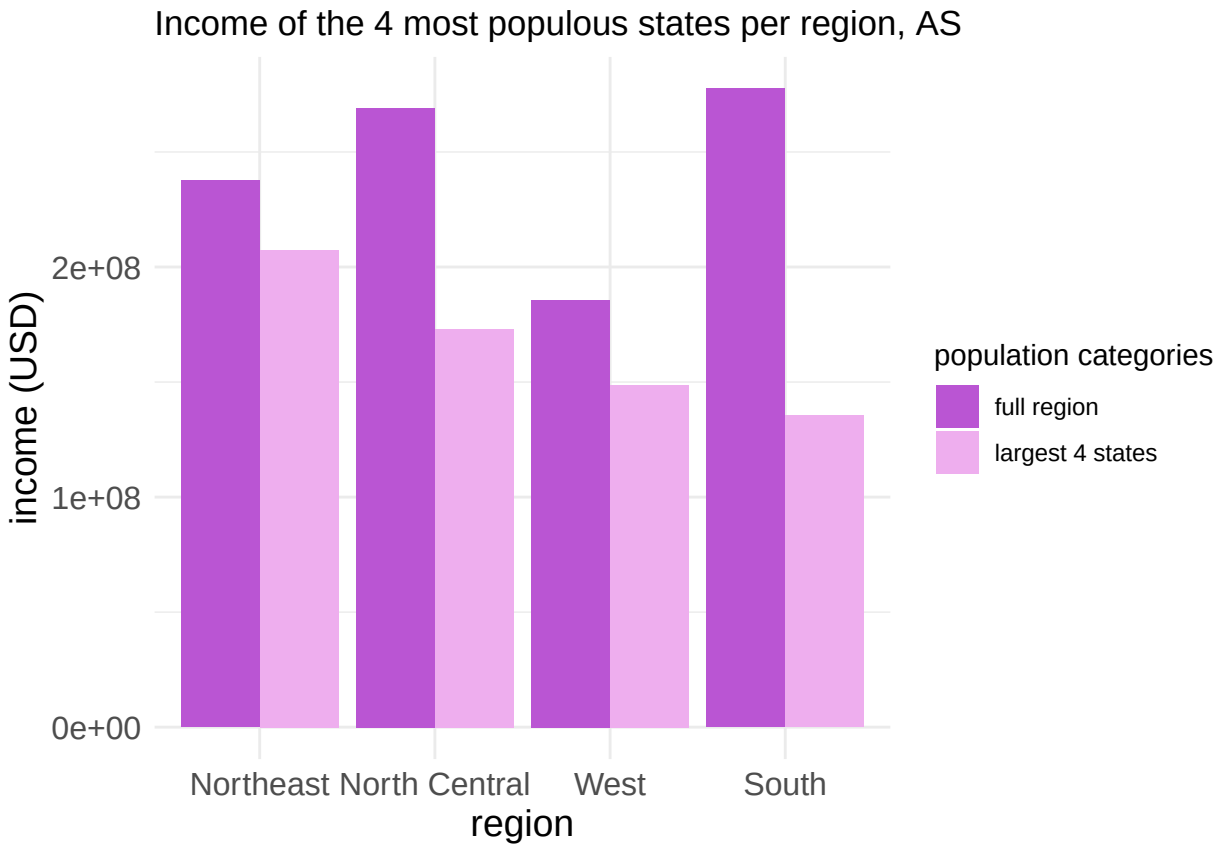


Q17*. Improve your plot from Q16 in the following ways:

- Instead of using *x=region*, use *x=fct_reorder2(region,income_category,income)*. Notice how the regions are now ordered by the length of one of the bars.
- Make the legend have a better title. You can do this inside the labs() command where you set the x label, y label, and title. Here you can use *fill="..."* to title the legend.
- Customize the colors by adding onto the plot a line of code like this (choose your own favorite color scheme!): *scale_fill_manual(labels=c("full region","largest 4 states"),values=c("gray","slateblue"))*

```
# copy and paste your Q16 code here, and add the modifications specified
ggplot(Q15,aes(x=fct_reorder2(region,income_category,income),y=income,fill=income_category)) +
    geom_col(position="dodge") +
    labs(title="Income of the 4 most populous states per region, AS",x="region",y="income (USD)", fill=
    theme_minimal() +
    larger_font +
```

```
    scale_fill_manual(labels=c("full region","largest 4 states"),values=c("mediumorchid","plum2"))
```



Q18*(response only). Interpret the plot by answering the following questions:

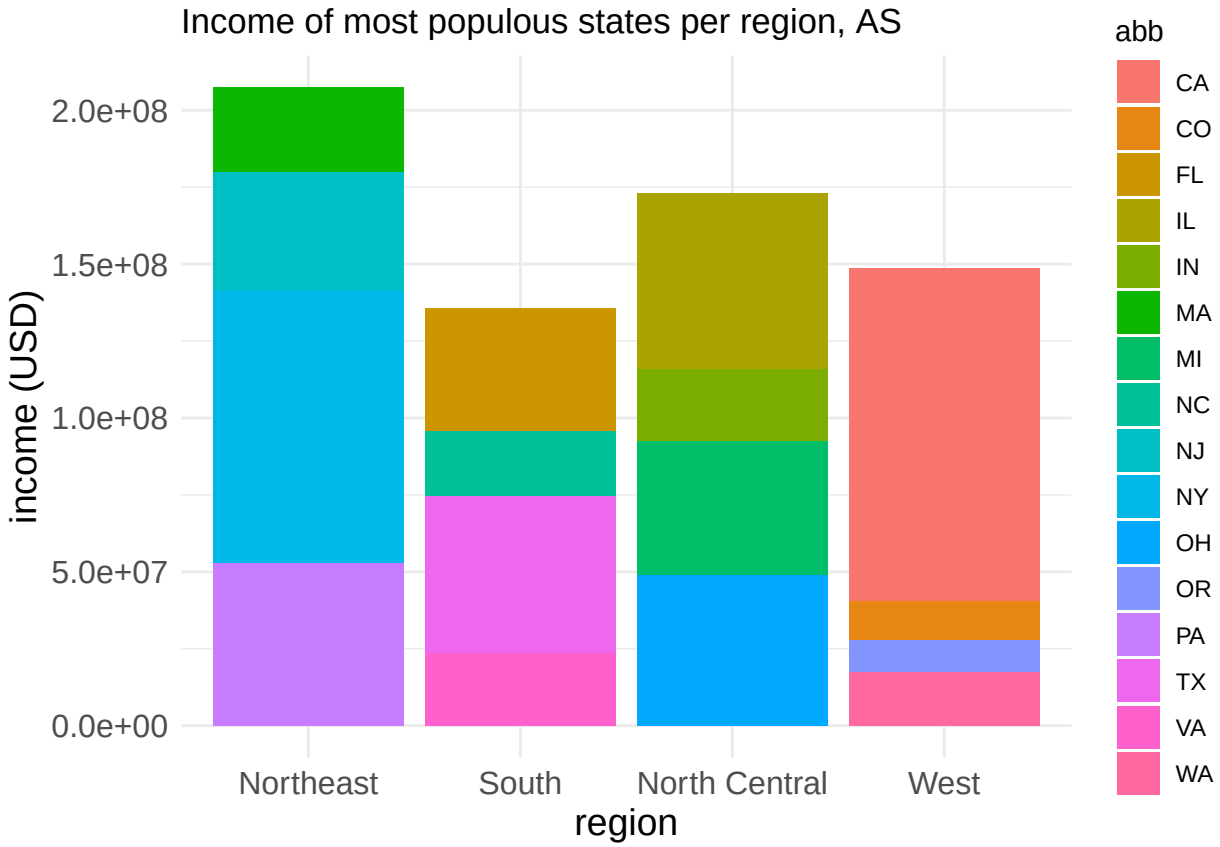
- In which two regions do the most populous states create the largest portion of the region's income? **Northeast and West**
- In which region is the income *least* determined by the most populous states? **South**

Q19*. Create a stacked bar plot for each region with the income of the 4 most populous states. We will improve the plot in several steps.

In this step:

- Create a stacked bar plot of income by region. Hint: start with the data frame from Q12.
- Encode the states using the fill color.
- Add title, labels, a theme, and enlarged font.

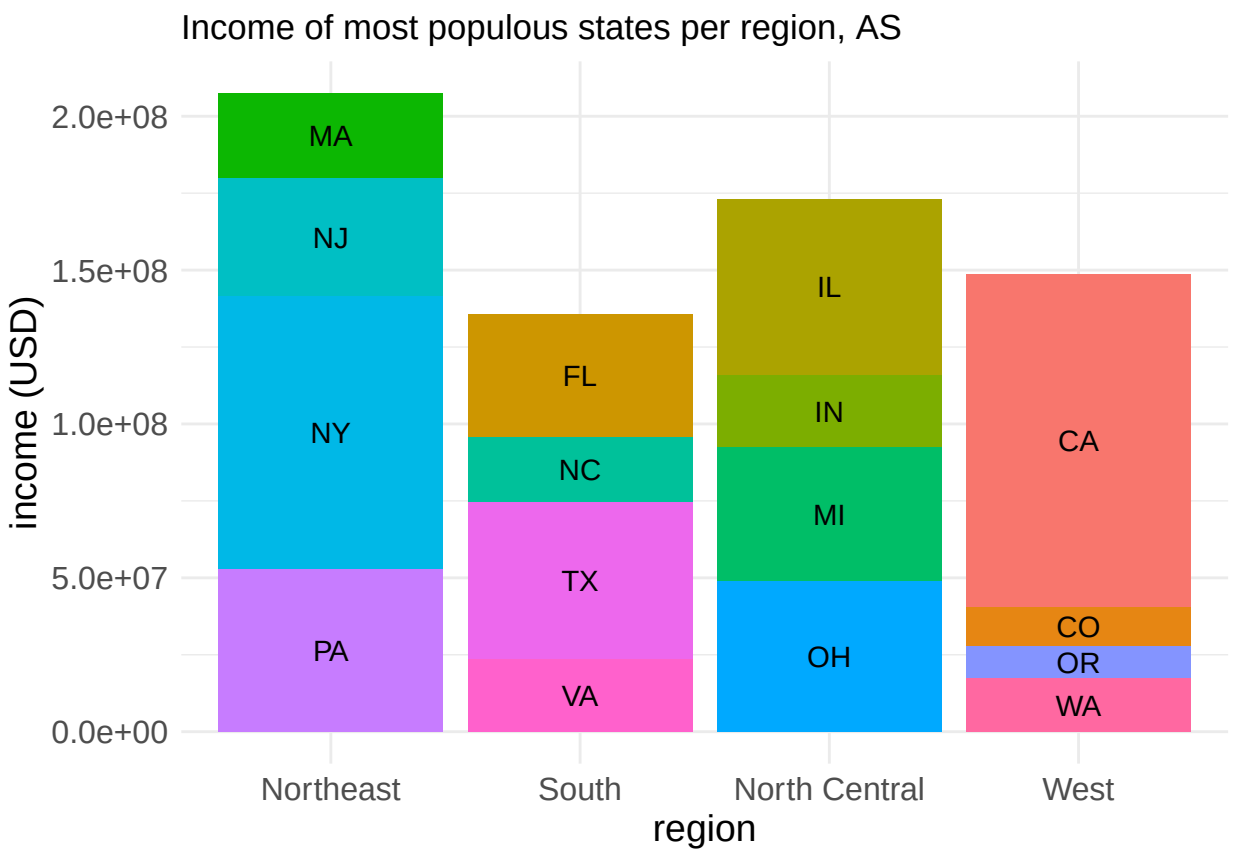
```
ggplot(Q12,aes(x=region,y=income,fill=abb)) +
    geom_col() +
    labs(title="Income of most populous states per region, AS",y="income (USD)") +
    theme_minimal() +
    larger_font
```



Q20*. Trying to match the color to the legend is too difficult. Improve the plot as follows:

- Add text to directly label the segments by adding the following code to the plot: *geom_text(aes(label=abb,y=income),positi*
- Get rid of the legend by putting *show.legend = FALSE* inside the geom_col() parentheses.

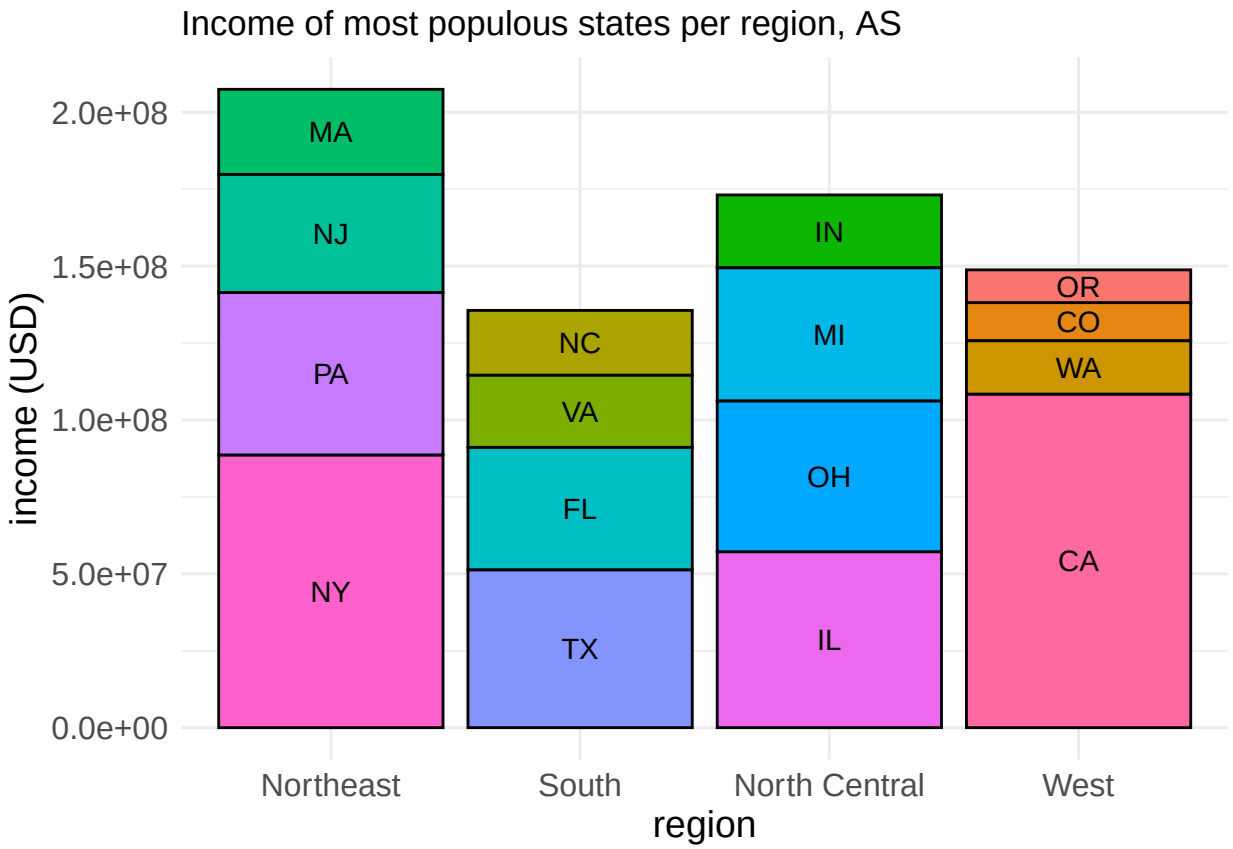
```
ggplot(Q12,aes(x=region,y=income,fill=abb)) +
    geom_col(show.legend = FALSE) +
    geom_text(aes(label=abb,y=income),position=position_stack(vjust=0.5)) +
    labs(title="Income of most populous states per region, AS",y="income (USD)") +
    theme_minimal() +
    larger_font
```



Q21*. The plot could still look better. Improve the plot as follows:

- Order the segments within each bar by replacing *fill=abb* with *fill=fct_reorder(abb,income)*.
- Add an outline around each segment by adding the following inside the parentheses for geom_col(): *color="black"*. Feel free to customize this color. Be sure to separate it from the *show.legend=FALSE* part using a comma.

```
ggplot(Q12,aes(x=region,y=income,fill=fct_reorder(abb,income))) +
    geom_col(show.legend = FALSE, color="black") +
    geom_text(aes(label=abb,y=income),position=position_stack(vjust=0.5)) +
    labs(title="Income of most populous states per region, AS",y="income (USD)") +
    theme_minimal() +
    larger_font
```



Q22*(response only). Interpret the plot by answering the following questions:

- Which two states make up a huge proportion of their region's income? **California and New York**
- Does this plot agree with the plot from Q17? Briefly explain your answer. **It does agree, but in the latter plot, we are looking at individual states per region. Versus in the Q17 plot, we examine top four states, as opposed to one, compared with the rest of the region.**

Part 3

- Main goal: See a case where data visualization gives us a **much** different insight than the summary statistics can.
- Secondary goal: Practice transforming data to make a richer plot than we could make directly from the raw data.

Q23. Run this to see what the ChickWeight dataset looks like:

```
ChickWeight %>% head()
```

```
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

```
ChickWeight %>% tail()
```

```
##     weight Time Chick Diet
## 573    155   12    50    4
## 574    175   14    50    4
## 575    205   16    50    4
## 576    234   18    50    4
## 577    264   20    50    4
## 578    264   21    50    4
```

Q24*. Create a table of the median chick weight for each diet. Call the column containing this median weight *median_weight*. Save this table as Q24.

```
Q24 <- ChickWeight %>%
    group_by(Diet) %>%
    summarize(median_weight = median(weight))
Q24
```

```
## # A tibble: 4 x 2
##   Diet  median_weight
##   <fct>         <dbl>
## 1 1              88
## 2 2             104.
## 3 3             126.
## 4 4             130.
```

We don't know if this tells the whole story. The chicks grew, but it's possible that those in one diet were measured less often than those in another. Let's look at some general exploratory analysis. Write code that answers these questions:

Q25*. How many rows are there for each diet? Save the result as Q25.

```
Q25 = table(ChickWeight$Diet)
Q25
```

```
##
##   1   2   3   4
## 220 120 120 118
```

Q26*. How many rows are there for each chick? Save the result as Q26.

Note! this will also tell you how many chicks there are in this data. There will be one row per chick.

```
Q26 <- ChickWeight %>%
   group_by(Chick) %>%
   summarize(Count = n())
   ### arrange()     *I'm trying to look up how to reorder Chick from 1-5 to make it cleaner, but having
Q26
```

```
## # A tibble: 50 x 2
##    Chick Count
##    <ord> <int>
##  1 18        2
##  2 16        7
##  3 15        8
##  4 13       12
##  5 9        12
##  6 20       12
```

```
##  7 10       12
##  8 8        11
##  9 17       12
## 10 19       12
## # i 40 more rows
```

Q27*. How many chicks were on each diet? Create a table that answers this question and save it to variable Q27.

Hint: Keep in mind the answers above, especially Q26.

```
Q27 <- ChickWeight %>%
   group_by(Diet) %>%
   summarize(count_distinct = n_distinct(Chick))
   ### tally()     *I'm also trying to figure out how to use the function tally(), instead of summarize
Q27
```

```
## # A tibble: 4 x 2
##   Diet  count_distinct
##   <fct>          <int>
## 1 1                 20
## 2 2                 10
## 3 3                 10
## 4 4                 10
```

Q28*. Plot the trajectory of the median weight for each diet over time.

This will require some of the functions we used earlier to get the data to the needed format prior to plotting. To do this, think about what format you would need for the data in order to have a time series plot for each diet. Work with the data until you get a data frame from the ChickWeight data frame that has the needed format. Then plot this result.

Keep in mind when making the plot:

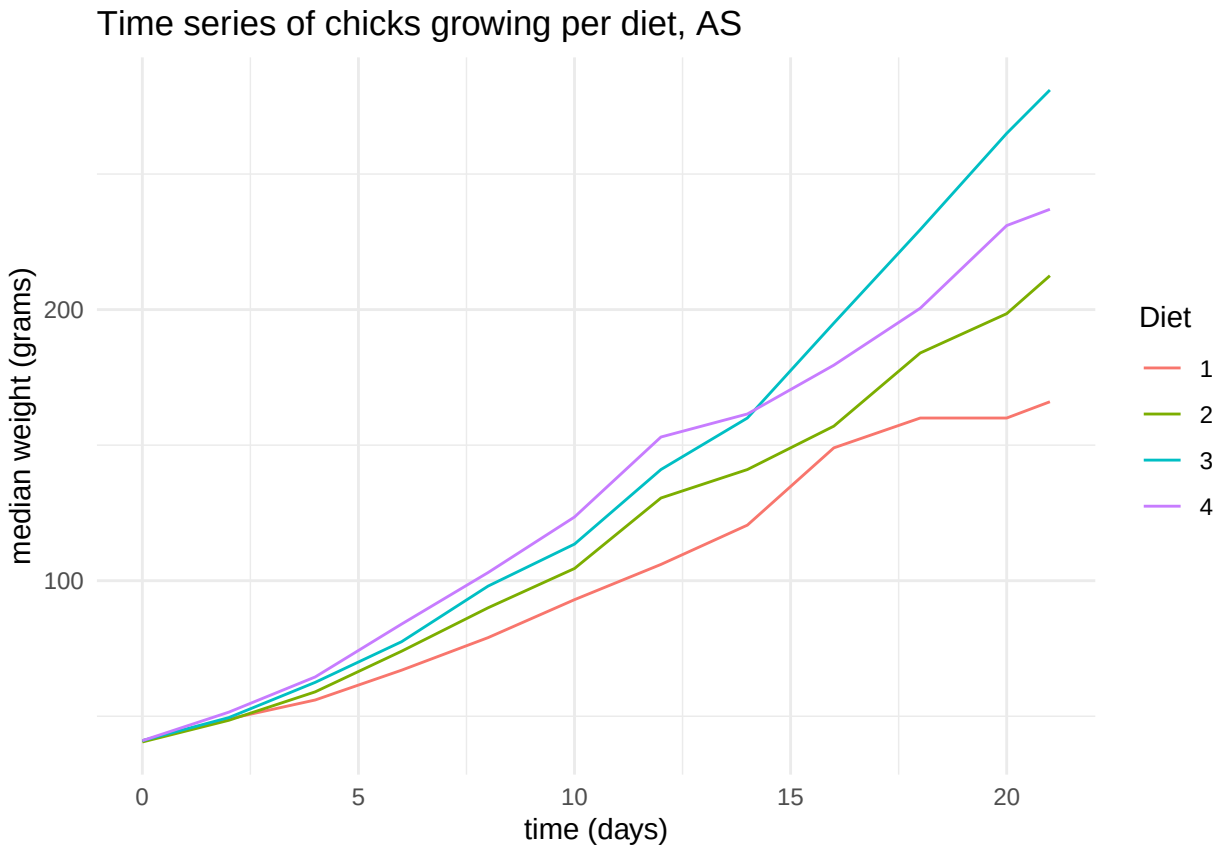
- Include units of grams and days as applicable.
- Include a title.
- Set a theme.

Hint: This plot should have smooth lines going up to the right (the chicks grow). If you get a plot that looks like zigzags or shark teeth, something went wrong - please take another look, and ask about this in the Slack channel.

```
Q28 <- ChickWeight %>%
   group_by(Time, Diet) %>%
   summarize(Q28 = median(weight))
```

```
## `summarise()` has grouped output by 'Time'. You can override using the
## `.groups` argument.
```

```
Q28 %>%
ggplot(aes(x=Time,y=Q28,color=Diet)) +
   geom_line() +
   labs(title="Time series of chicks growing per diet, AS",x="time (days)",y="median weight (grams)") +
   theme_minimal()
```



Q29*(response only). Draw a conclusion from this plot that answers the following:

- Which diet is best for chick growth?
- Did the medians for each diet in Q24 tell the whole story, or did the plot give a better understanding, and why?
- How did a visualization give an advantage over summary statistics (i.e. the median) in this case?

- If we consider the maximum growth, then Diet 3 has the most impressive trend upwards. - The time series plot tells a more in-depth story over the summary in Q24. We see the entire lifespan of the chicks by diet, being 0-21 days. - Similarly to what was stated previously, we see the lifespan over the x-axis which shows linear data. We also see the median weight over time per diet of the chicks, with the actual weights (not median) ranging from 35-373. This is more information than we would get from a orrelation coefficient figure for each different diet.