The Sixth R of Data Analysis

Reliable, **R**epresentative, **R**eplicable, **R**eady **R**eporting

Chris Sewell, Director

Troy Jordan & Josh Marhevka, Deputy Directors

David Schmidt, Chief Economist

Prepared by the Research & Analysis Bureau

DETR
Nevada Department of Employment,
Training and Rehabilitation

NevadaWorkforce.com

Research & Analysis | Nevada's premier source of workforce and economic information and analysis

What do I hope to address?

Introduction

What is R and why should I use it?

Application

Making a table and a chart, the easy and repeatable way.

Integration and Iteration

Making charts and tables is a good start, but now it's time to scale up.

Performance

I love using R and I now want it to do EVERYTHING! How do I make it go faster?

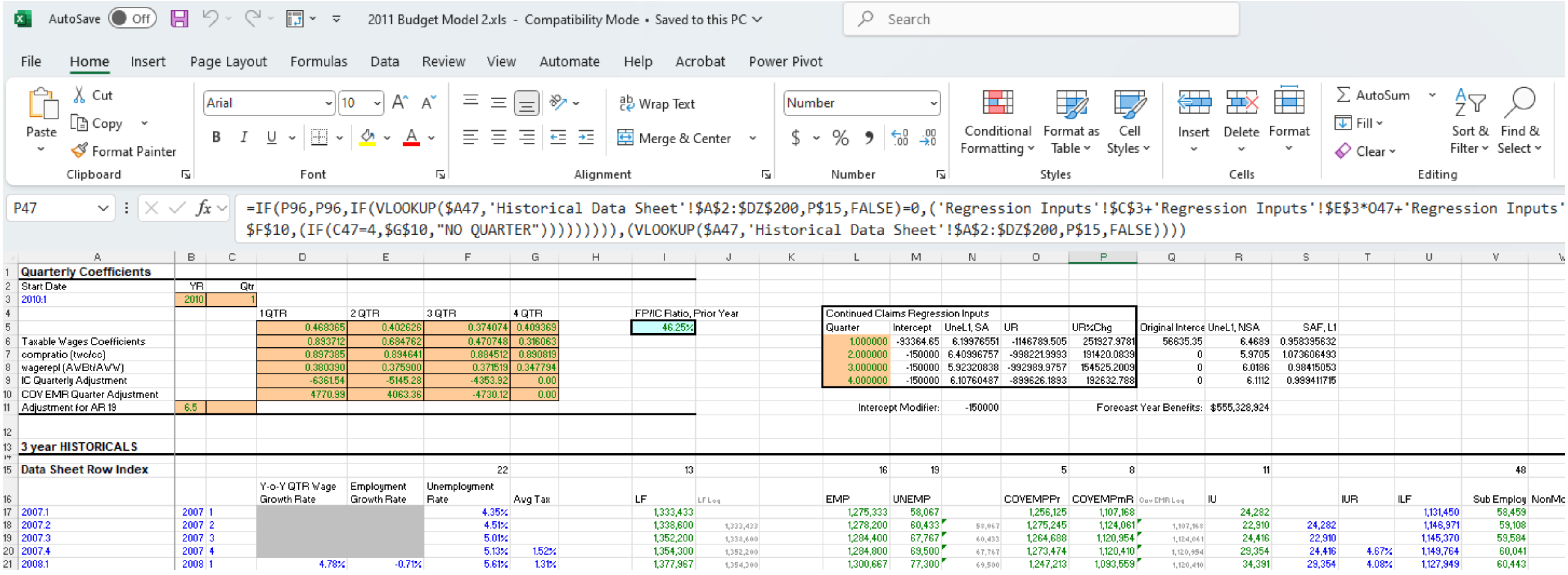
Standing on the shoulders of giants…

R for Data Science
https://r4ds.hadley.nz/
https://www.tidyverse.org/

Kyle Walker and Tidycensus
https://walker-data.com/census-r/
https://bsky.app/profile/kylewalker.bsky.social

Forecasting Principles and Practice
https://otexts.com/fpp3/
https://tidyverts.org/

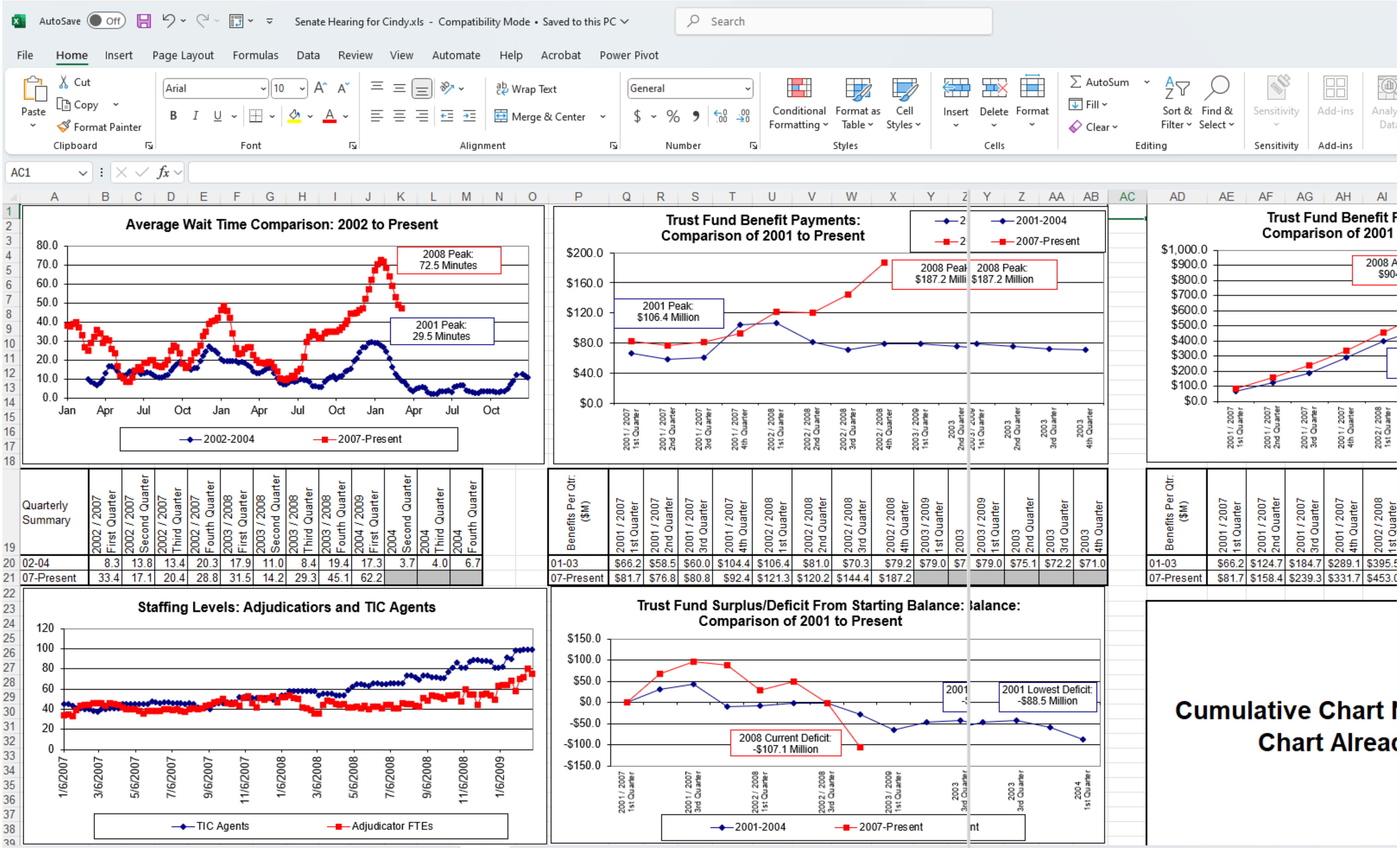
Once Upon a Time...



P47

```
=IF(P96,P96,IF(VLOOKUP($A47,'Historical Data Sheet'!$A$2:$DZ$200,P$15,FALSE)=0,('Regression Inputs'!$C$3+'Regression Inputs'!$E$3*O47+'Regression Inputs'
$F$10,(IF(C47=4,$G$10,"NO QUARTER"))))))))),(VLOOKUP($A47,'Historical Data Sheet'!$A$2:$DZ$200,P$15,FALSE))))
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	**Quarterly Coefficients**																						
2	Start Date	YR	Qtr																				
3	2010:1	2010	1																				
4				1QTR	2 QTR	3 QTR	4 QTR		FP/IC Ratio, Prior Year			Continued Claims Regression Inputs											
5				0.468365	0.402626	0.374074	0.409369		46.25%			Quarter	Intercept	UneL1, SA	UR	UR%Chg	Original Interce	UneL1, NSA	SAF, L1				
6	Taxable Wages Coefficients			0.893712	0.684762	0.470748	0.316063					1.000000	-93364.65	6.19976551	-1146789.505	251927.9781	56635.35	6.4689	0.958395632				
7	compratio (twc/cc)			0.897385	0.894641	0.884512	0.890819					2.000000	-150000	6.40996757	-998221.9993	191420.0839	0	5.9705	1.073606493				
8	wagerepl (AWBt/AWW)			0.380390	0.375900	0.371519	0.347794					3.000000	-150000	5.92320838	-992989.9757	154525.2009	0	6.0186	0.98415053				
9	IC Quarterly Adjustment			-6361.54	-5145.28	-4353.92	0.00					4.000000	-150000	6.10760487	-899626.1893	192632.788	0	6.1112	0.999411715				
10	COV EMR Quarter Adjustment			4770.99	4063.36	-4730.12	0.00																
11	Adjustment for AR 19	6.5											Intercept Modifier:	-150000			Forecast Year Benefits:	$555,328,924					
12																							
13	**3 year HISTORICALS**																						
14																							
15	**Data Sheet Row Index**					22			13			16	19		5	8		11				48	
16				Y-o-Y QTR Wage Growth Rate	Employment Growth Rate	Unemployment Rate	Avg Tax		LF	LF Lag		EMP	UNEMP		COVEMPPr	COVEMPmR	Cov EMR Lag	IU		IUR	ILF	Sub Employ	NonMc
17	2007.1	2007	1			4.35%			1,333,433			1,275,333	58,067		1,256,125	1,107,168		24,282			1,131,450	58,459	
18	2007.2	2007	2			4.51%			1,338,600	1,333,433		1,278,200	60,433	58,067	1,275,245	1,124,061	1,107,168	22,910	24,282		1,146,971	59,108	
19	2007.3	2007	3			5.01%			1,352,200	1,338,600		1,284,400	67,767	60,433	1,264,688	1,120,954	1,124,061	24,416	22,910		1,145,370	59,584	
20	2007.4	2007	4			5.13%	1.52%		1,354,300	1,352,200		1,284,800	69,500	67,767	1,273,474	1,120,410	1,120,954	29,354	24,416	4.67%	1,149,764	60,041	
21	2008.1	2008	1	4.78%	-0.71%	5.61%	1.31%		1,377,967	1,354,300		1,300,667	77,300	69,500	1,247,213	1,093,559	1,120,410	34,391	29,354	4.08%	1,127,949	60,443	



Quarterly Summary	2002 / 2007 First Quarter	2002 / 2007 Second Quarter	2002 / 2007 Third Quarter	2002 / 2007 Fourth Quarter	2003 / 2008 First Quarter	2003 / 2008 Second Quarter	2003 / 2008 Third Quarter	2003 / 2008 Fourth Quarter	2004 / 2009 First Quarter	2004 Second Quarter	2004 Third Quarter	2004 Fourth Quarter
02-04	8.3	13.8	13.4	20.3	17.9	11.0	8.4	19.4	17.3	3.7	4.0	6.7
07-Present	33.4	17.1	20.4	28.8	31.5	14.2	29.3	45.1	62.2			

Benefits Per Qtr: ($M)	2001 / 2007 1st Quarter	2001 / 2007 2nd Quarter	2001 / 2007 3rd Quarter	2001 / 2007 4th Quarter	2002 / 2008 1st Quarter	2002 / 2008 2nd Quarter	2002 / 2008 3rd Quarter	2002 / 2008 4th Quarter	2003 / 2009 1st Quarter	2003	2003 / 2009 1st Quarter	2003 2nd Quarter	2003 3rd Quarter	2003 4th Quarter
01-03	$66.2	$58.5	$60.0	$104.4	$106.4	$81.0	$70.3	$79.2	$79.0	$7	$79.0	$75.1	$72.2	$71.0
07-Present	$81.7	$76.8	$80.8	$92.4	$121.3	$120.2	$144.4	$187.2						

Benefits Per Qtr: ($M)	2001 / 2007 1st Quarter	2001 / 2007 2nd Quarter	2001 / 2007 3rd Quarter	2001 / 2007 4th Quarter	2002 / 2008 1st Quarter
01-03	$66.2	$124.7	$184.7	$289.1	$395.5
07-Present	$81.7	$158.4	$239.3	$331.7	$453.0

```
=IF(P96,P96,IF(VLOOKUP($A47,'Historical Data
Sheet'!$A$2:$DZ$200,P$15,FALSE)=0,('Regressi
on Inputs'!$C$3+'Regression
Inputs'!$E$3*O47+'Regression
Inputs'!$G$3*B47+(IF(C47=1,$D$10,(IF(C47=2,$
E$10,(IF(C47=3,$F$10,(IF(C47=4,$G$10,"NO
QUARTER")))))))))),(VLOOKUP($A47,'Historical
Data Sheet'!$A$2:$DZ$200,P$15,FALSE))))
```

VLOOKUP

SUMIFS

INDEX(MATCH)

IFERROR

{Array Formulas}

Why Excel?

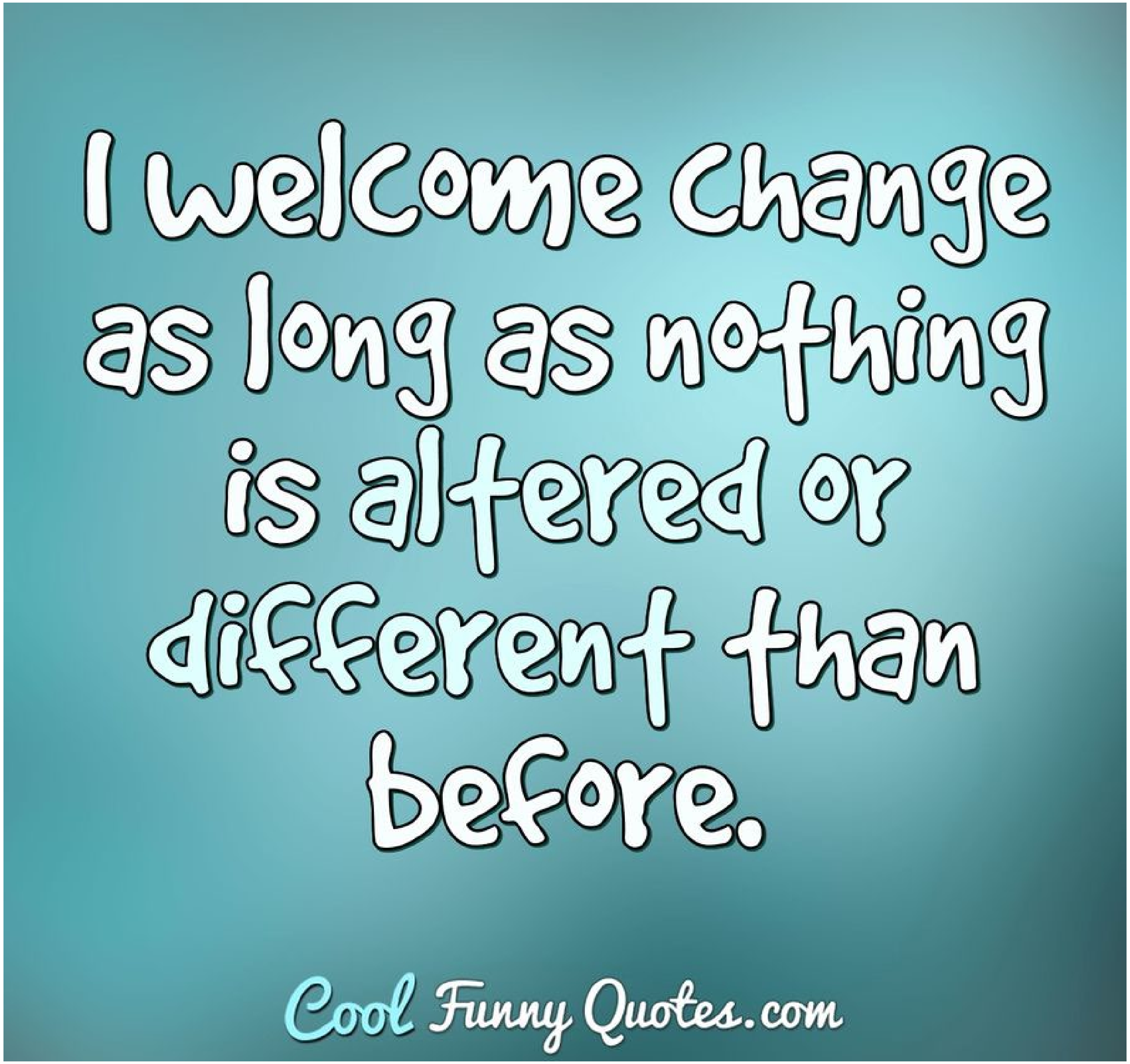
Availability

Price

Power

Familiarity

Flexibility



I welcome change as long as nothing is altered or different than before.

Cool Funny Quotes.com

Excel		R
☑	Availability	☑
☑	Price	☑
	Power	☑
☑	Familiarity	
	Flexibility	☑

What is it?

A highly extensible language and environment for statistical programming and graphics.

https://www.r-project.org/about.html

R Studio: software that provides additional ease-of-use in interacting with R.

https://posit.co/download/rstudio-desktop/

“Posit, RStudio, and Shiny are trademarks of Posit Software, PBC, all rights reserved, and may be registered in the United States Patent and Trademark Office and in other countries.”

What makes R so powerful?

Packages are bundles of functions that build on the basic functionality of R to streamline workflow and integrate processes.

Packages take the foundation of R and build a huge variety of amazing options.

CRAN provides a free repository of packages which meet certain requirements in a central location.

Anyone can build their own packages, too!

https://cran.r-project.org/

Recommended Packages

tidyverse - Bundle of workflow-related packages

tidycensus - Easy access to Census Bureau data

tigris - Easy access to Census Bureau shapefiles

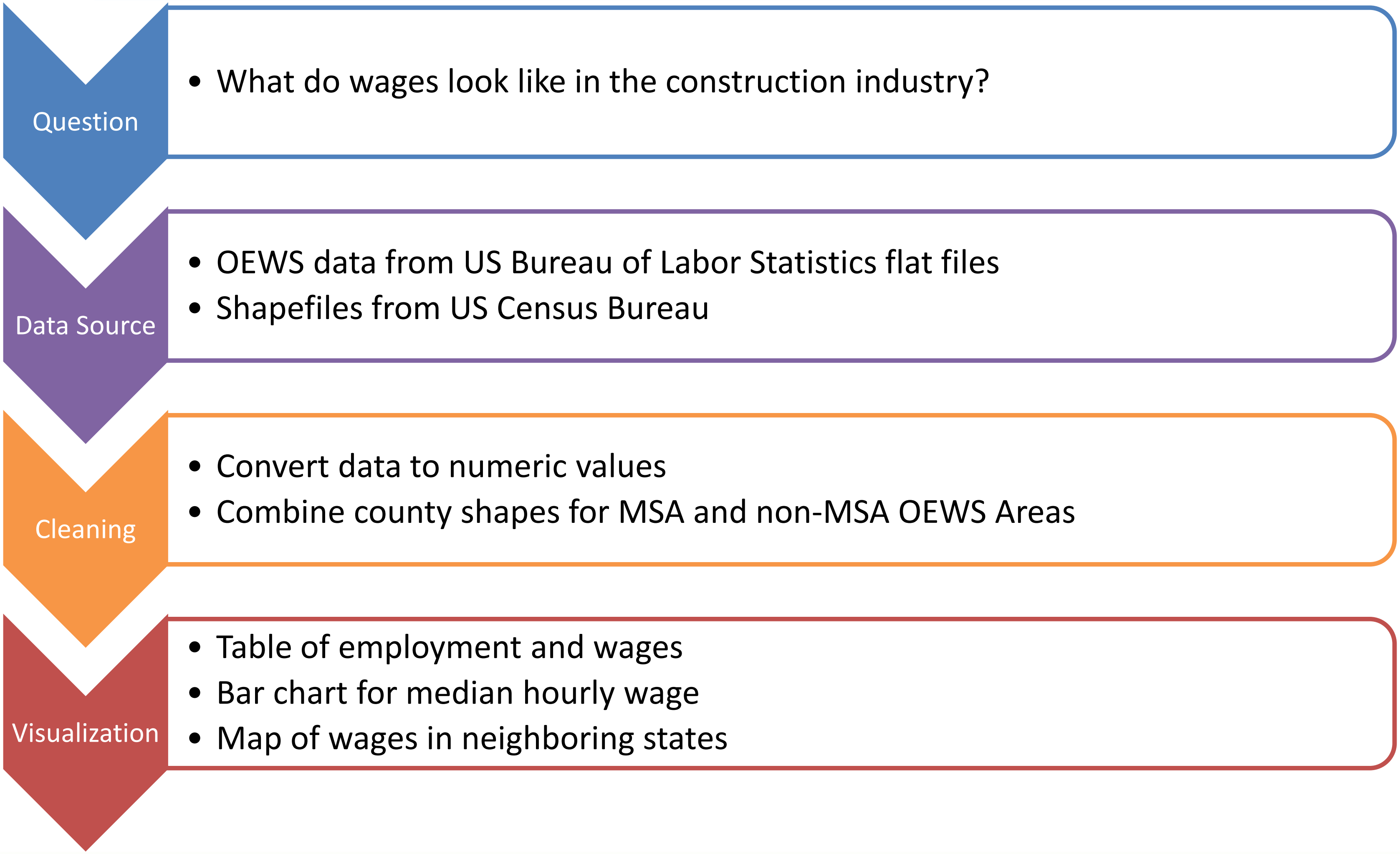
sf - Tools for working with spatial data

tsibble - Working with time series data

openxlsx - Building Excel workbooks

Data Analysis Workflow

Code at: https://github.com/schmidtDETR/OEWS-Mapping-and-Visualization



Question

- What do wages look like in the construction industry?

Data Source

- OEWS data from US Bureau of Labor Statistics flat files
- Shapefiles from US Census Bureau

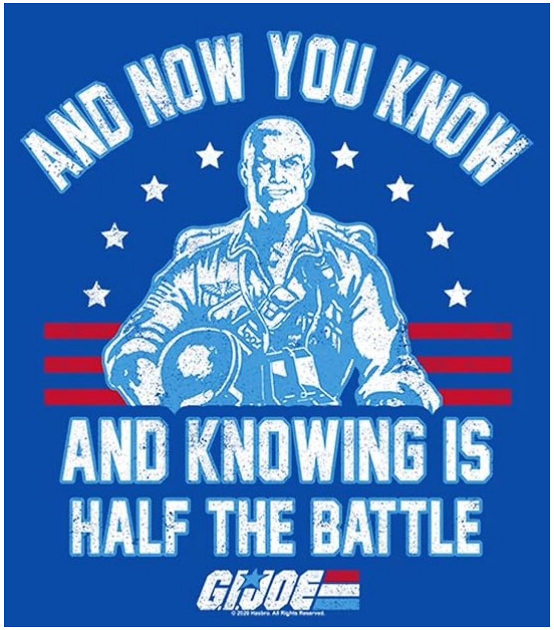
Cleaning

- Convert data to numeric values
- Combine county shapes for MSA and non-MSA OEWS Areas

Visualization

- Table of employment and wages
- Bar chart for median hourly wage
- Map of wages in neighboring states

Data Sources – Your Superpower



BLS Flat Files: https://download.bls.gov/pub/time.series/
This has almost all the core BLS data, except QCEW.

FRED: https://fred.stlouisfed.org/
A massive reserve of economic indicators, *all* of which can be pulled straight into R using the tidyquant package.

LODES: https://lehd.ces.census.gov/data/#lodes
Flat files for LEHD origin-destination data.

CPS Data: https://www.census.gov/data/datasets/time-series/demo/cps/cps-basic.html
Raw CPS survey data.

UI Data Downloads: https://oui.doleta.gov/unemploy/DataDownloads.asp
Flat file downloads, data maps, and report instructions for all UI reports.

QCEW Slices: https://www.bls.gov/cew/additional-resources/open-data/csv-data-slices.htm
This gives you access to BLS QCEW data for all published areas.

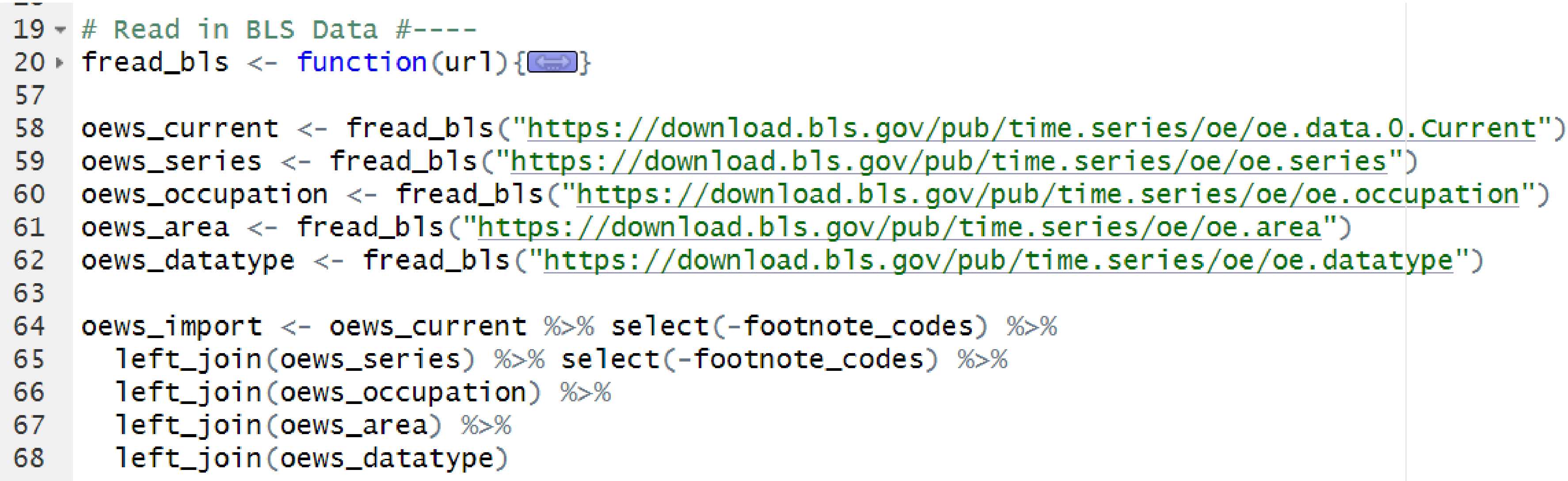
Alternative Measures: https://www.bls.gov/lau/stalt-moave.xlsx
Excel file with historical data for BLS alternative measures of labor underutilization for states.

Getting the Data in R

Data Source

Getting OEWS data from BLS – you can download the files manually, or you can use a script in R to go straight to the source.

I use a function so that I can eliminate manual steps in the process. The computer can do it, so I don’t want to waste time being a bad computer.

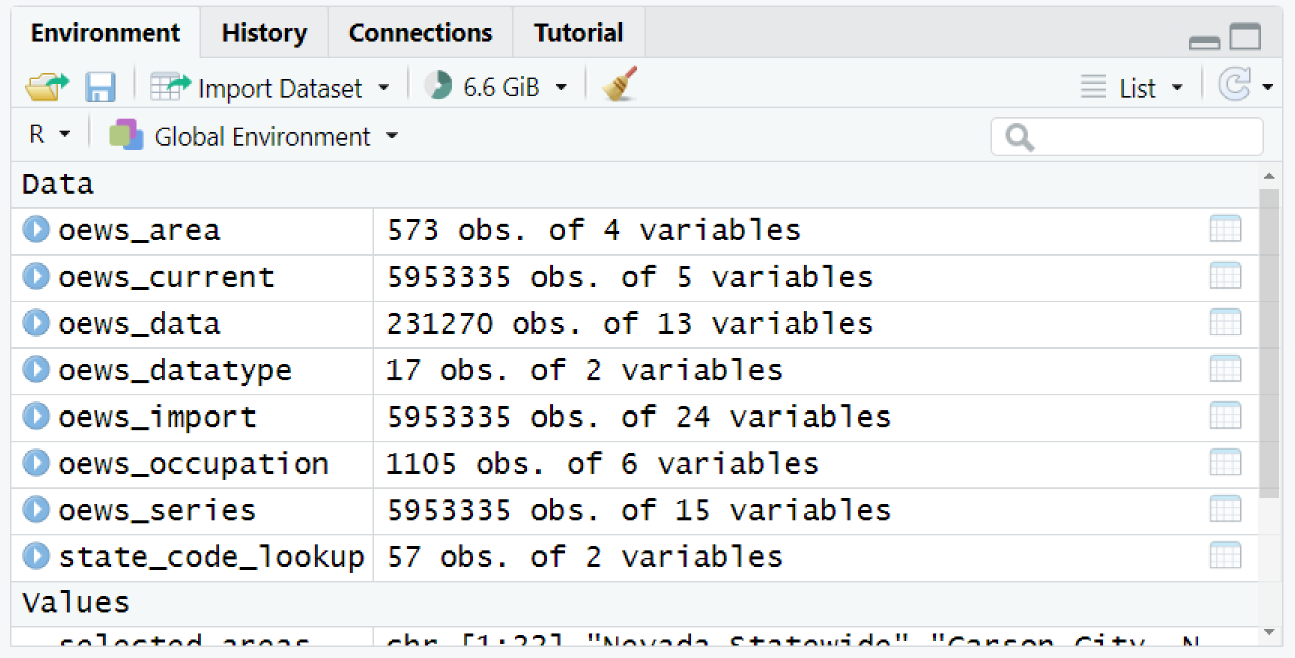


```
# Read in BLS Data #----
fread_bls <- function(url){⟺}

oews_current <- fread_bls("https://download.bls.gov/pub/time.series/oe/oe.data.0.Current")
oews_series <- fread_bls("https://download.bls.gov/pub/time.series/oe/oe.series")
oews_occupation <- fread_bls("https://download.bls.gov/pub/time.series/oe/oe.occupation")
oews_area <- fread_bls("https://download.bls.gov/pub/time.series/oe/oe.area")
oews_datatype <- fread_bls("https://download.bls.gov/pub/time.series/oe/oe.datatype")

oews_import <- oews_current %>% select(-footnote_codes) %>%
  left_join(oews_series) %>% select(-footnote_codes) %>%
  left_join(oews_occupation) %>%
  left_join(oews_area) %>%
  left_join(oews_datatype)
```

After downloading the data, I join together the flat files, which are set up like a relational database.

Getting the Data in R

Data Source



Data	
oews_area	573 obs. of 4 variables
oews_current	5953335 obs. of 5 variables
oews_data	231270 obs. of 13 variables
oews_datatype	17 obs. of 2 variables
oews_import	5953335 obs. of 24 variables
oews_occupation	1105 obs. of 6 variables
oews_series	5953335 obs. of 15 variables
state_code_lookup	57 obs. of 2 variables
Values	
selected_areas	chr [1:22] "Nevada Statewide" "Carson City, N

After I load the data, it appears in my “Environment” – the working space for data I have available to me.

6 million rows of data, all loaded, joined, and cleaned in 20 seconds. Sorry, Excel. You’ve just been left in the dust.

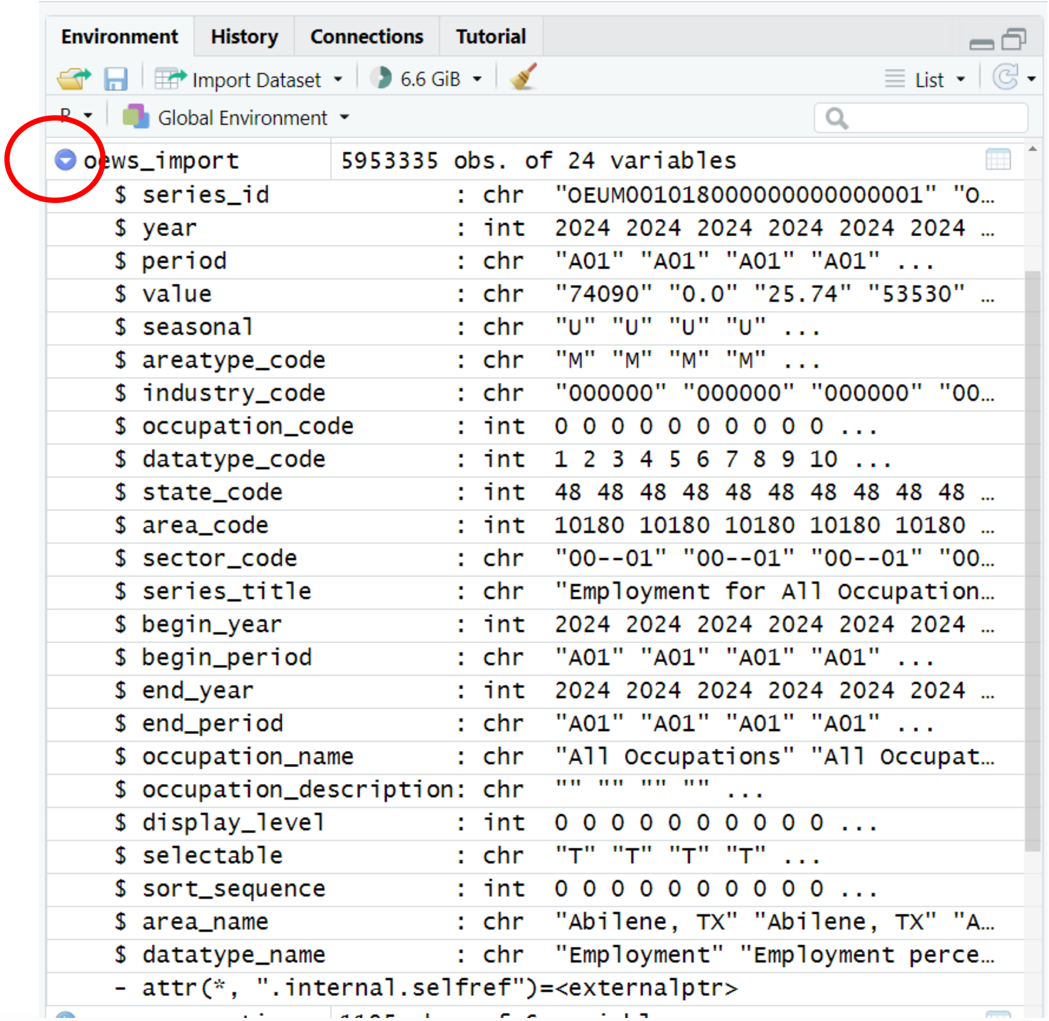
Getting the Data in R

Data Source

Looking at my joined data frame, I can see the **columns** (usually denoted with $ in R), as well as their data types.

As with any database-like structure, paying attention to data types can mean success or failure.

Knowing what we have, let's start cleaning this up!



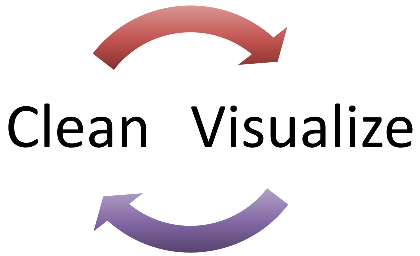
```
oews_import            5953335 obs. of 24 variables
    $ series_id            : chr  "OEUM001018000000000000001" "O…
    $ year                 : int  2024 2024 2024 2024 2024 2024 …
    $ period               : chr  "A01" "A01" "A01" "A01" ...
    $ value                : chr  "74090" "0.0" "25.74" "53530" …
    $ seasonal             : chr  "U" "U" "U" "U" ...
    $ areatype_code        : chr  "M" "M" "M" "M" ...
    $ industry_code        : chr  "000000" "000000" "000000" "00…
    $ occupation_code      : int  0 0 0 0 0 0 0 0 0 0 ...
    $ datatype_code        : int  1 2 3 4 5 6 7 8 9 10 ...
    $ state_code           : int  48 48 48 48 48 48 48 48 48 48 …
    $ area_code            : int  10180 10180 10180 10180 10180 …
    $ sector_code          : chr  "00--01" "00--01" "00--01" "00…
    $ series_title         : chr  "Employment for All Occupation…
    $ begin_year           : int  2024 2024 2024 2024 2024 2024 …
    $ begin_period         : chr  "A01" "A01" "A01" "A01" ...
    $ end_year             : int  2024 2024 2024 2024 2024 2024 …
    $ end_period           : chr  "A01" "A01" "A01" "A01" ...
    $ occupation_name      : chr  "All Occupations" "All Occupat…
    $ occupation_description: chr  "" "" "" "" ...
    $ display_level        : int  0 0 0 0 0 0 0 0 0 0 ...
    $ selectable           : chr  "T" "T" "T" "T" ...
    $ sort_sequence        : int  0 0 0 0 0 0 0 0 0 0 ...
    $ area_name            : chr  "Abilene, TX" "Abilene, TX" "A…
    $ datatype_name        : chr  "Employment" "Employment perce…
    - attr(*, ".internal.selfref")=<externalptr>
```

Cleaning the Data in R

Cleaning

I want to get this data ready to answer my questions: I want to know about employment and wages for selected occupations in and around Nevada.

1. I need my data values to be numeric, so that I can graph them like a number.
2. I want my different data types to be in columns, with the values below the data name.
3. I want to be able to sort the data in a way that shows Nevada first and other states second (for emphasis), and that shows statewide and then substate areas (for clarity).
4. I want some friendly names like “Nevada”, not “32”


Clean Visualize

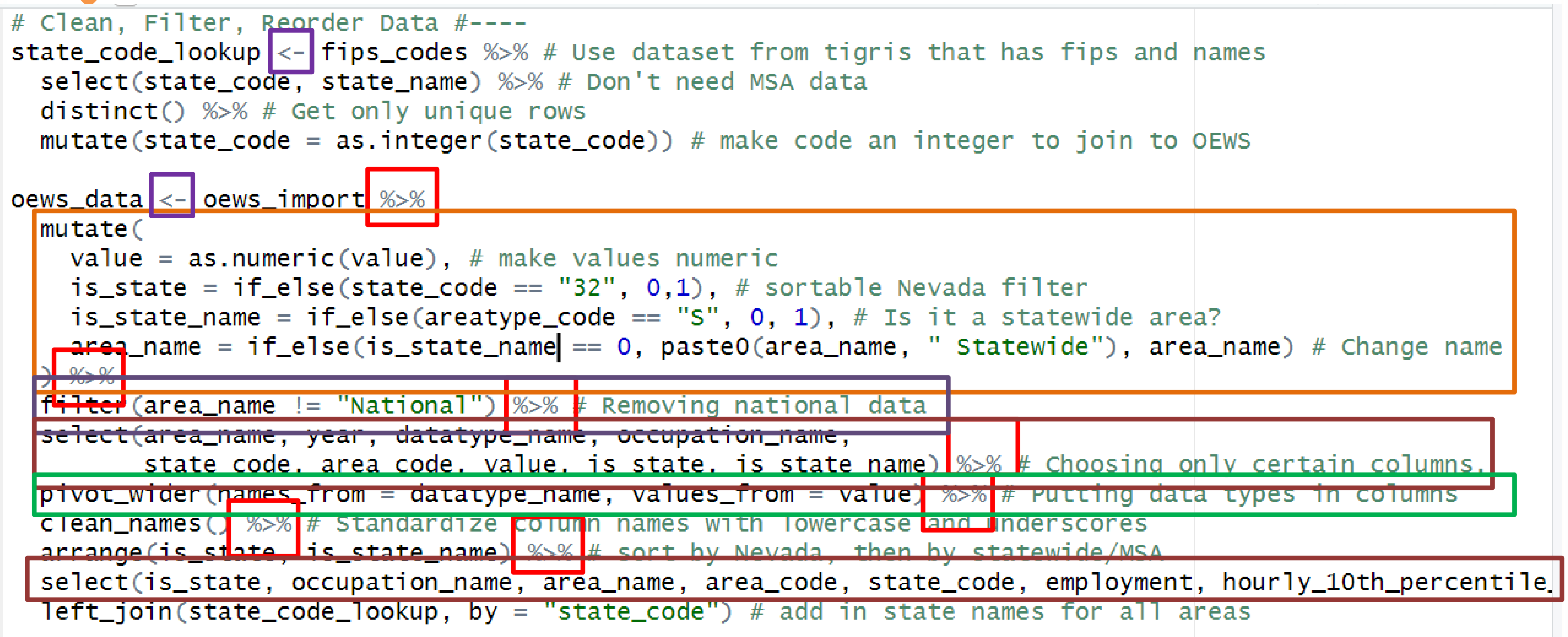
This is an iterative process!

Cleaning the Data in R

Cleaning

Walking through a few key steps using tidyverse functions:

`<-`, `%>%`, `mutate`, `filter`, `select`, `pivot_wider`

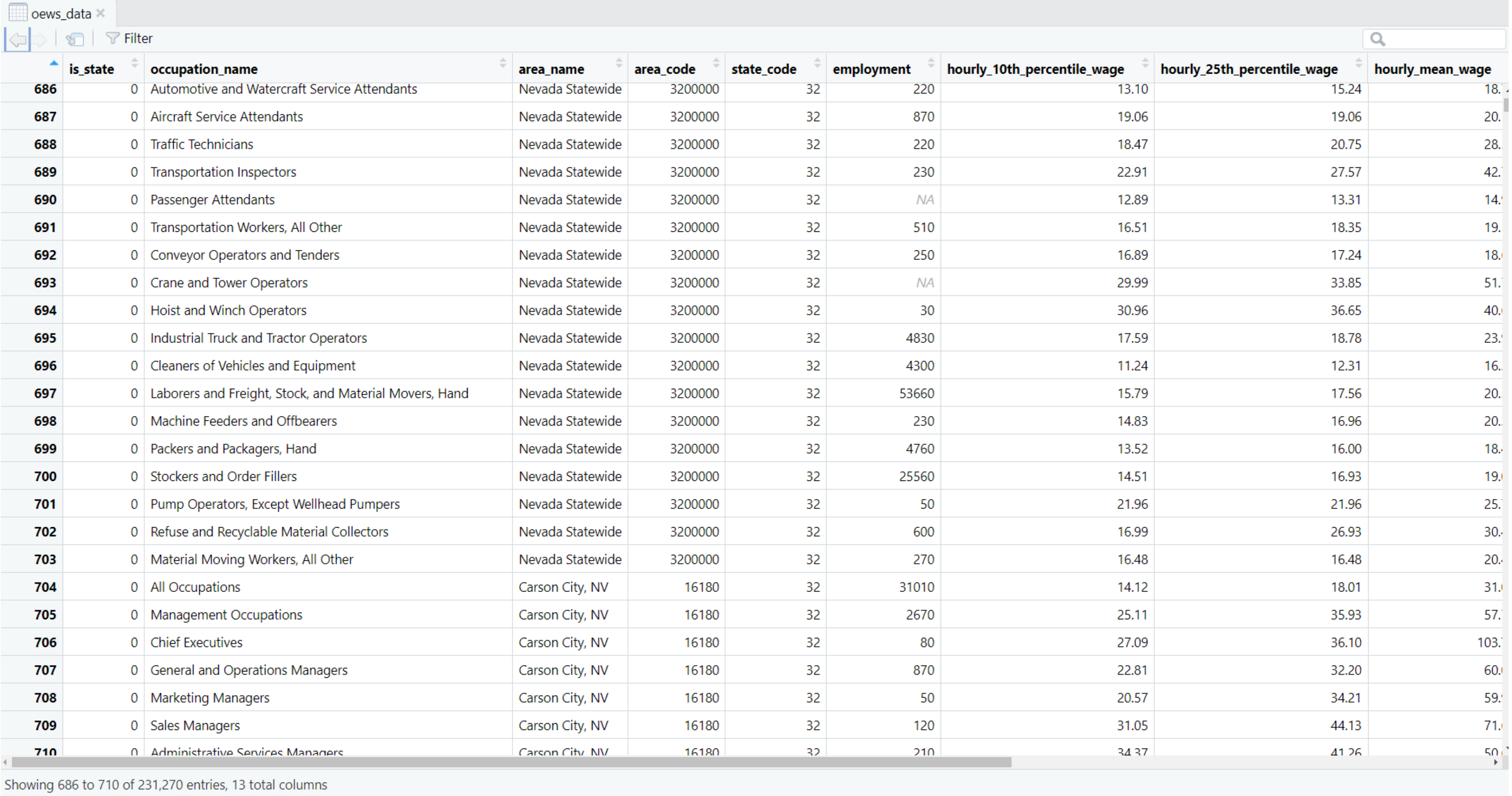


```
# Clean, Filter, Reorder Data #----
state_code_lookup <- fips_codes %>% # Use dataset from tigris that has fips and names
  select(state_code, state_name) %>% # Don't need MSA data
  distinct() %>% # Get only unique rows
  mutate(state_code = as.integer(state_code)) # make code an integer to join to OEWS

oews_data <- oews_import %>%
  mutate(
    value = as.numeric(value), # make values numeric
    is_state = if_else(state_code == "32", 0,1), # sortable Nevada filter
    is_state_name = if_else(areatype_code == "S", 0, 1), # Is it a statewide area?
    area_name = if_else(is_state_name == 0, paste0(area_name, " Statewide"), area_name) # Change name
  ) %>%
  filter(area_name != "National") %>% # Removing national data
  select(area_name, year, datatype_name, occupation_name,
         state_code, area_code, value, is_state, is_state_name) %>% # Choosing only certain columns.
  pivot_wider(names_from = datatype_name, values_from = value) %>% # Putting data types in columns
  clean_names() %>% # Standardize column names with lowercase and underscores
  arrange(is_state, is_state_name) %>% # sort by Nevada, then by statewide/MSA
  select(is_state, occupation_name, area_name, area_code, state_code, employment, hourly_10th_percentile,
  left_join(state_code_lookup, by = "state_code") # add in state names for all areas
```

Cleaning

Cleaning the Data in R

A peak at what we have available now… looking good!



	is_state	occupation_name	area_name	area_code	state_code	employment	hourly_10th_percentile_wage	hourly_25th_percentile_wage	hourly_mean_wage
686	0	Automotive and Watercraft Service Attendants	Nevada Statewide	3200000	32	220	13.10	15.24	18.
687	0	Aircraft Service Attendants	Nevada Statewide	3200000	32	870	19.06	19.06	20.
688	0	Traffic Technicians	Nevada Statewide	3200000	32	220	18.47	20.75	28.
689	0	Transportation Inspectors	Nevada Statewide	3200000	32	230	22.91	27.57	42.
690	0	Passenger Attendants	Nevada Statewide	3200000	32	NA	12.89	13.31	14.
691	0	Transportation Workers, All Other	Nevada Statewide	3200000	32	510	16.51	18.35	19.
692	0	Conveyor Operators and Tenders	Nevada Statewide	3200000	32	250	16.89	17.24	18.
693	0	Crane and Tower Operators	Nevada Statewide	3200000	32	NA	29.99	33.85	51.
694	0	Hoist and Winch Operators	Nevada Statewide	3200000	32	30	30.96	36.65	40.
695	0	Industrial Truck and Tractor Operators	Nevada Statewide	3200000	32	4830	17.59	18.78	23.
696	0	Cleaners of Vehicles and Equipment	Nevada Statewide	3200000	32	4300	11.24	12.31	16.
697	0	Laborers and Freight, Stock, and Material Movers, Hand	Nevada Statewide	3200000	32	53660	15.79	17.56	20.
698	0	Machine Feeders and Offbearers	Nevada Statewide	3200000	32	230	14.83	16.96	20.
699	0	Packers and Packagers, Hand	Nevada Statewide	3200000	32	4760	13.52	16.00	18.
700	0	Stockers and Order Fillers	Nevada Statewide	3200000	32	25560	14.51	16.93	19.
701	0	Pump Operators, Except Wellhead Pumpers	Nevada Statewide	3200000	32	50	21.96	21.96	25.
702	0	Refuse and Recyclable Material Collectors	Nevada Statewide	3200000	32	600	16.99	26.93	30.
703	0	Material Moving Workers, All Other	Nevada Statewide	3200000	32	270	16.48	16.48	20.
704	0	All Occupations	Carson City, NV	16180	32	31010	14.12	18.01	31.
705	0	Management Occupations	Carson City, NV	16180	32	2670	25.11	35.93	57.
706	0	Chief Executives	Carson City, NV	16180	32	80	27.09	36.10	103.
707	0	General and Operations Managers	Carson City, NV	16180	32	870	22.81	32.20	60.
708	0	Marketing Managers	Carson City, NV	16180	32	50	20.57	34.21	59.
709	0	Sales Managers	Carson City, NV	16180	32	120	31.05	44.13	71.
710	0	Administrative Services Managers	Carson City, NV	16180	32	210	34.37	41.26	50.

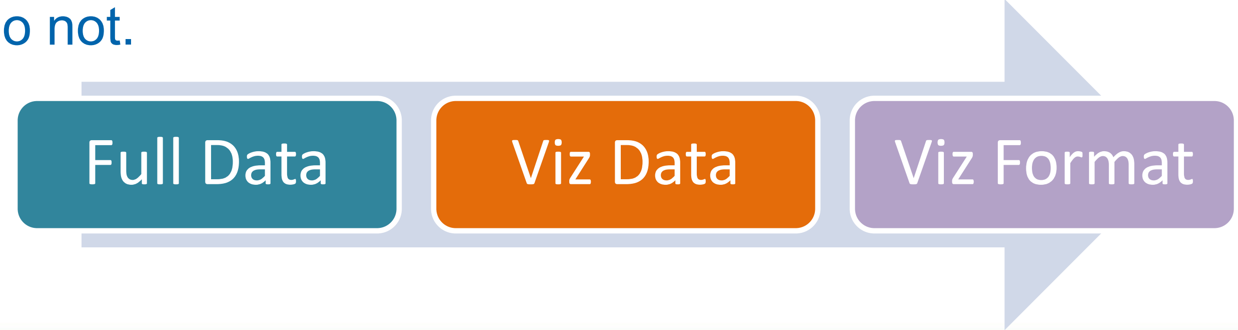
Showing 686 to 710 of 231,270 entries, 13 total columns

Visualizing the Data in R

Visualization

Our data already looks kind of like the table that we want, it's just too large to display in a pretty way (still 231,270 rows!). Now we'll want to subset our data to display what we want.

1. I want a table, to highlight some hard numbers for different data elements.
2. I want a bar chart highlighting wages within different states and how they vary.
3. I want a map, to highlight regional proximity in a way that separate bar charts do not.



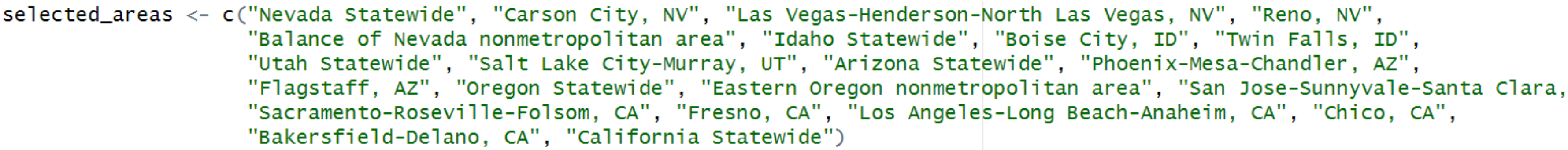
Visualizing the Data in R

Visualization

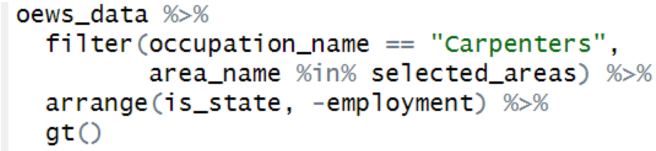
1. I want a table, to highlight some hard numbers for different data elements.

I like to use the **gt** package for tables. https://gt.rstudio.com/

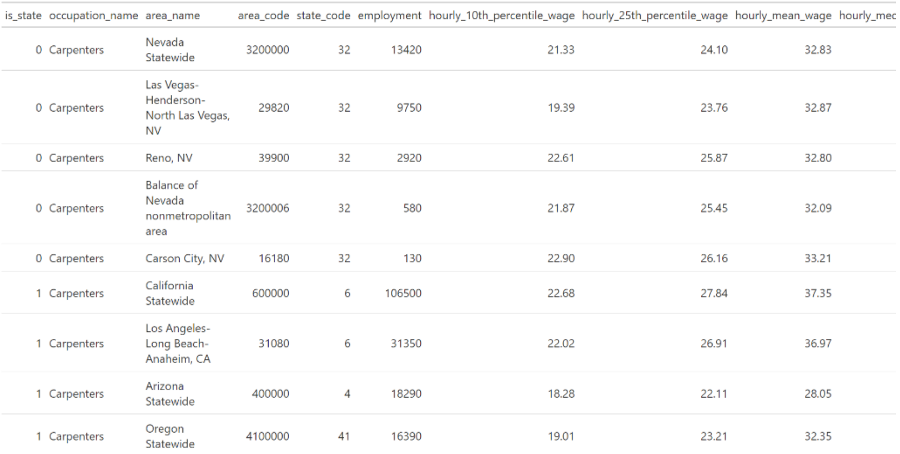
First, I'm spelling out what areas I want to include in my table.



```
selected_areas <- c("Nevada Statewide", "Carson City, NV", "Las Vegas-Henderson-North Las Vegas, NV", "Reno, NV",
                    "Balance of Nevada nonmetropolitan area", "Idaho Statewide", "Boise City, ID", "Twin Falls, ID",
                    "Utah Statewide", "Salt Lake City-Murray, UT", "Arizona Statewide", "Phoenix-Mesa-Chandler, AZ",
                    "Flagstaff, AZ", "Oregon Statewide", "Eastern Oregon nonmetropolitan area", "San Jose-Sunnyvale-Santa Clara,
                    "Sacramento-Roseville-Folsom, CA", "Fresno, CA", "Los Angeles-Long Beach-Anaheim, CA", "Chico, CA",
                    "Bakersfield-Delano, CA", "California Statewide")
```



```
oews_data %>%
  filter(occupation_name == "Carpenters",
         area_name %in% selected_areas) %>%
  arrange(is_state, -employment) %>%
  gt()
```



is_state	occupation_name	area_name	area_code	state_code	employment	hourly_10th_percentile_wage	hourly_25th_percentile_wage	hourly_mean_wage	hourly_mec
0	Carpenters	Nevada Statewide	3200000	32	13420	21.33	24.10	32.83	
0	Carpenters	Las Vegas-Henderson-North Las Vegas, NV	29820	32	9750	19.39	23.76	32.87	
0	Carpenters	Reno, NV	39900	32	2920	22.61	25.87	32.80	
0	Carpenters	Balance of Nevada nonmetropolitan area	3200006	32	580	21.87	25.45	32.09	
0	Carpenters	Carson City, NV	16180	32	130	22.90	26.16	33.21	
1	Carpenters	California Statewide	600000	6	106500	22.68	27.84	37.35	
1	Carpenters	Los Angeles-Long Beach-Anaheim, CA	31080	6	31350	22.02	26.91	36.97	
1	Carpenters	Arizona Statewide	400000	4	18290	18.28	22.11	28.05	
1	Carpenters	Oregon Statewide	4100000	41	16390	19.01	23.21	32.35	

The basic table is simple, but just spits out the data the way it looks right now. To group it and make it pretty, we want to do a bit more.

Visualizing the Data in R

Visualization

Add a group name for each state to separate them visually.

Hide columns we don't need to show.

Format numbers with $ and , marks.

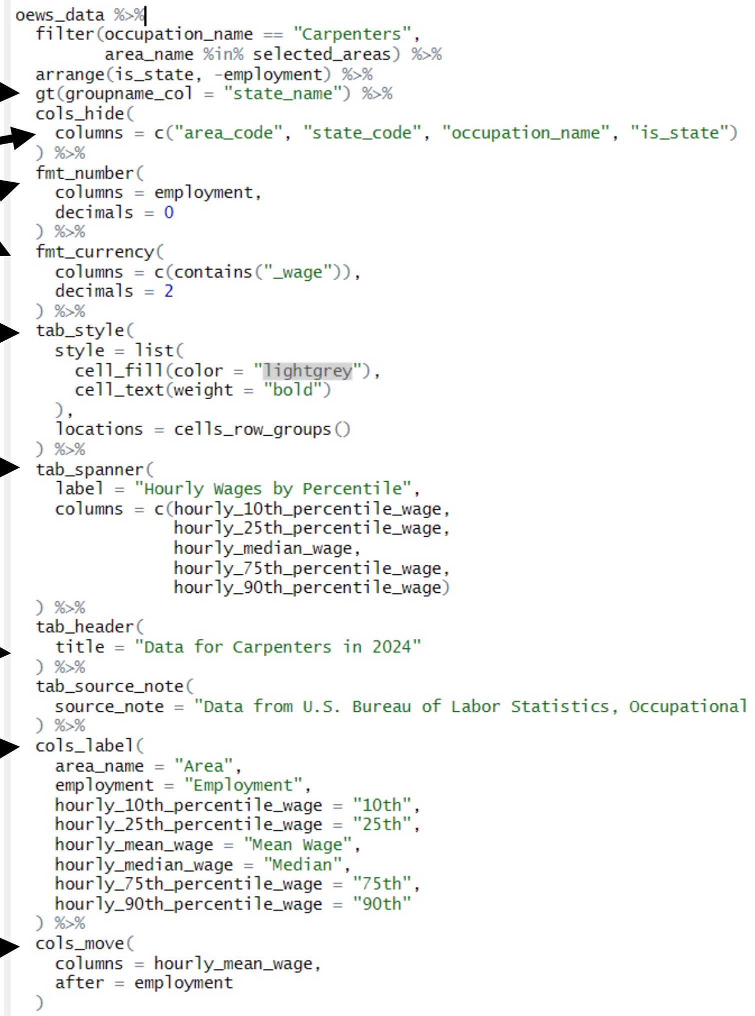
Format cells for state group names.

Add a group header for wage percentiles.

Add a title and footnote.

Rename columns with user-friendly labels

Move mean wage next to employment data



```r
oews_data %>%
  filter(occupation_name == "Carpenters",
         area_name %in% selected_areas) %>%
  arrange(is_state, -employment) %>%
  gt(groupname_col = "state_name") %>%
  cols_hide(
    columns = c("area_code", "state_code", "occupation_name", "is_state")
  ) %>%
  fmt_number(
    columns = employment,
    decimals = 0
  ) %>%
  fmt_currency(
    columns = c(contains("_wage")),
    decimals = 2
  ) %>%
  tab_style(
    style = list(
      cell_fill(color = "lightgrey"),
      cell_text(weight = "bold")
    ),
    locations = cells_row_groups()
  ) %>%
  tab_spanner(
    label = "Hourly Wages by Percentile",
    columns = c(hourly_10th_percentile_wage,
                hourly_25th_percentile_wage,
                hourly_median_wage,
                hourly_75th_percentile_wage,
                hourly_90th_percentile_wage)
  ) %>%
  tab_header(
    title = "Data for Carpenters in 2024"
  ) %>%
  tab_source_note(
    source_note = "Data from U.S. Bureau of Labor Statistics, Occupational
  ) %>%
  cols_label(
    area_name = "Area",
    employment = "Employment",
    hourly_10th_percentile_wage = "10th",
    hourly_25th_percentile_wage = "25th",
    hourly_mean_wage = "Mean Wage",
    hourly_median_wage = "Median",
    hourly_75th_percentile_wage = "75th",
    hourly_90th_percentile_wage = "90th"
  ) %>%
  cols_move(
    columns = hourly_mean_wage,
    after = employment
  )
```

Visualizing the Data in R

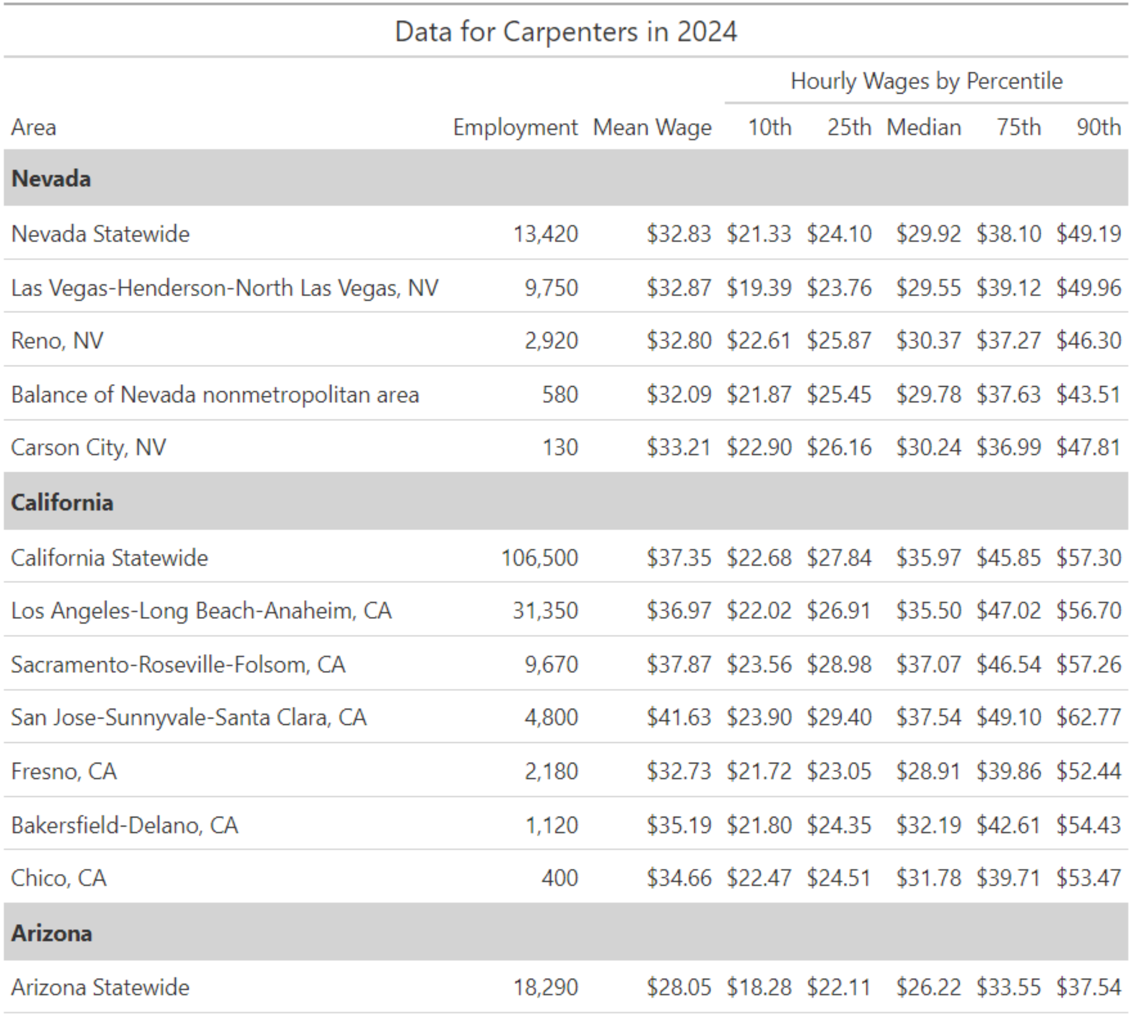
Visualization

Formatting makes all the difference!

Remaining challenges:

1. Getting this to the recipient.
2. Scaling this up (we hard coded “Carpenters” as a filter and in the title).

We’ll come back to #2 soon!



Data for Carpenters in 2024

			Hourly Wages by Percentile				
Area	Employment	Mean Wage	10th	25th	Median	75th	90th
Nevada							
Nevada Statewide	13,420	$32.83	$21.33	$24.10	$29.92	$38.10	$49.19
Las Vegas-Henderson-North Las Vegas, NV	9,750	$32.87	$19.39	$23.76	$29.55	$39.12	$49.96
Reno, NV	2,920	$32.80	$22.61	$25.87	$30.37	$37.27	$46.30
Balance of Nevada nonmetropolitan area	580	$32.09	$21.87	$25.45	$29.78	$37.63	$43.51
Carson City, NV	130	$33.21	$22.90	$26.16	$30.24	$36.99	$47.81
California							
California Statewide	106,500	$37.35	$22.68	$27.84	$35.97	$45.85	$57.30
Los Angeles-Long Beach-Anaheim, CA	31,350	$36.97	$22.02	$26.91	$35.50	$47.02	$56.70
Sacramento-Roseville-Folsom, CA	9,670	$37.87	$23.56	$28.98	$37.07	$46.54	$57.26
San Jose-Sunnyvale-Santa Clara, CA	4,800	$41.63	$23.90	$29.40	$37.54	$49.10	$62.77
Fresno, CA	2,180	$32.73	$21.72	$23.05	$28.91	$39.86	$52.44
Bakersfield-Delano, CA	1,120	$35.19	$21.80	$24.35	$32.19	$42.61	$54.43
Chico, CA	400	$34.66	$22.47	$24.51	$31.78	$39.71	$53.47
Arizona							
Arizona Statewide	18,290	$28.05	$18.28	$22.11	$26.22	$33.55	$37.54

Visualizing the Data in R

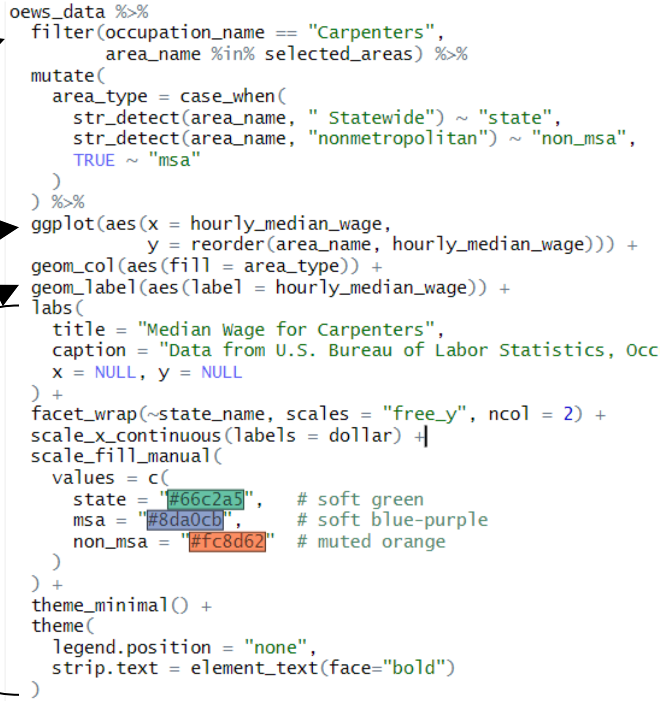
Visualization

2. I want a bar chart highlighting wages within different states and how they vary.
3. I want a map, to highlight regional proximity in a way that separate bar charts do not.

I will use ggplot2 to do each of these. https://ggplot2.tidyverse.org/

Elements of a ggplot:

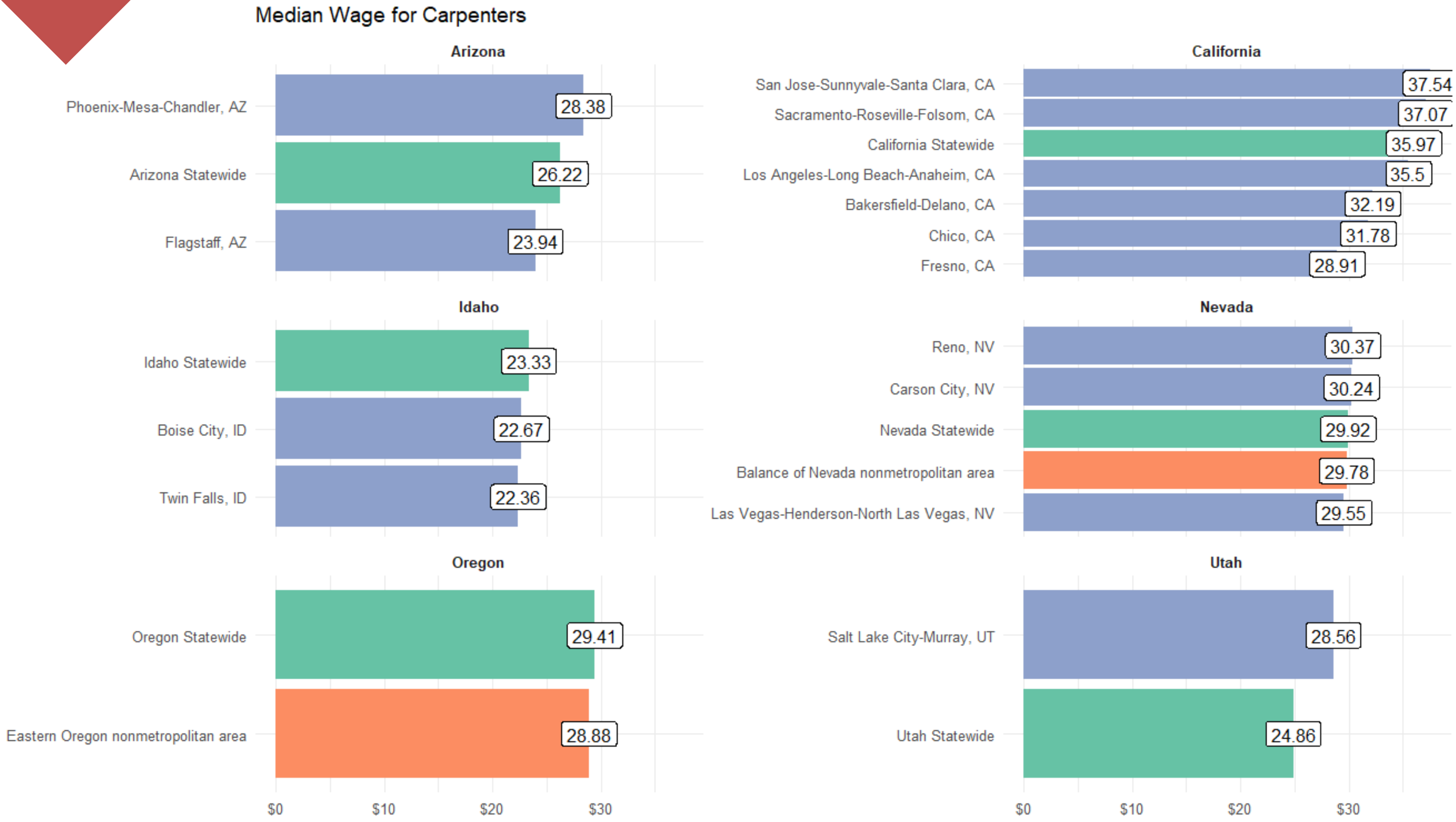
1. Data set – what are you plotting?
2. Mapping (aesthetics) – what data columns will define your dynamic variables – x, y, color, fil, size, etc? Uses aes()
3. Geometries – what shape will that mapped data take? Uses geom_()
4. Scales, Labels, and Beautification



```
oews_data %>%
  filter(occupation_name == "Carpenters",
         area_name %in% selected_areas) %>%
  mutate(
    area_type = case_when(
      str_detect(area_name, " Statewide") ~ "state",
      str_detect(area_name, "nonmetropolitan") ~ "non_msa",
      TRUE ~ "msa"
    )
  ) %>%
  ggplot(aes(x = hourly_median_wage,
             y = reorder(area_name, hourly_median_wage))) +
  geom_col(aes(fill = area_type)) +
  geom_label(aes(label = hourly_median_wage)) +
  labs(
    title = "Median Wage for Carpenters",
    caption = "Data from U.S. Bureau of Labor Statistics, Occ
    x = NULL, y = NULL
  ) +
  facet_wrap(~state_name, scales = "free_y", ncol = 2) +
  scale_x_continuous(labels = dollar) +
  scale_fill_manual(
    values = c(
      state = "#66c2a5",    # soft green
      msa = "#8da0cb",      # soft blue-purple
      non_msa = "#fc8d62"   # muted orange
    )
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    strip.text = element_text(face="bold")
  )
```

Visualization

Visualizing the Data in R



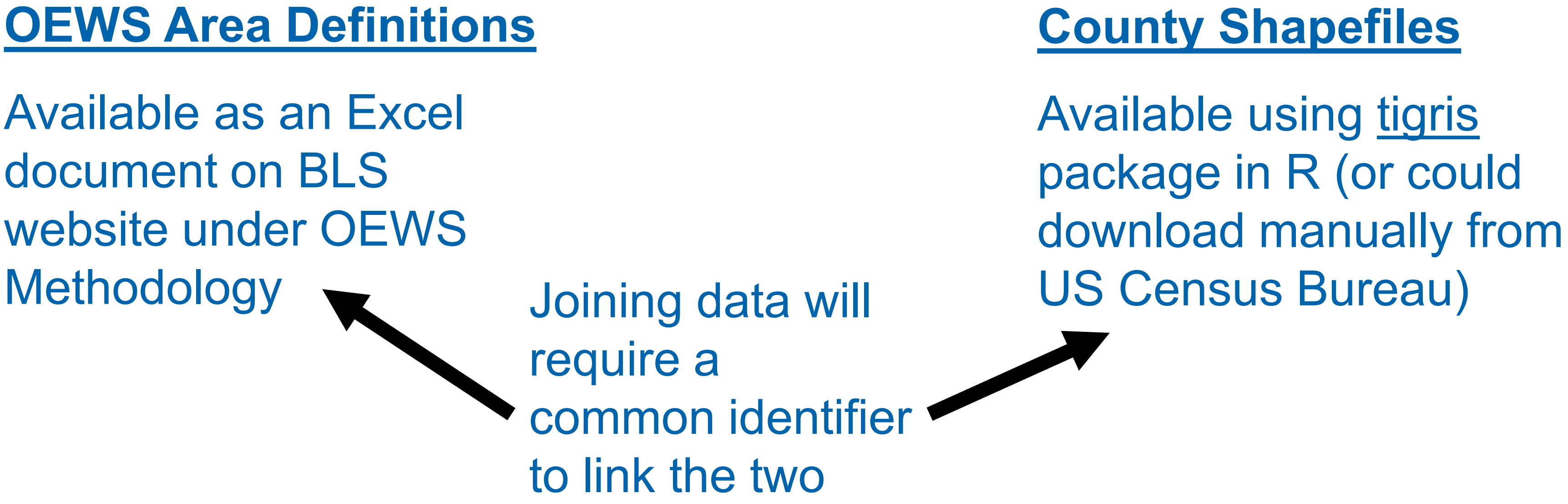
Data from U.S. Bureau of Labor Statistics, Occupational Employment and Wage Statistics

Visualizing the Data in R

Visualization

3. I want a map, to highlight regional proximity in a way that separate bar charts do not.

To do a map, I need to get information about the geographic shapes I want to plot. OEWS does not produce data for all counties, but its data is built from counties. I need to start with county data and map those counties to the OEWS area definitions.



OEWS Area Definitions

Available as an Excel document on BLS website under OEWS Methodology

County Shapefiles

Available using tigris package in R (or could download manually from US Census Bureau)

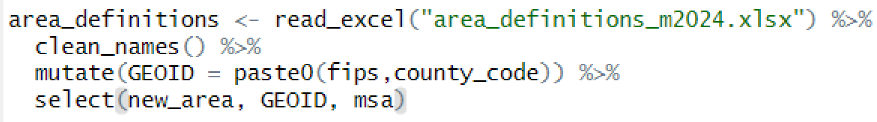
Joining data will require a common identifier to link the two

Visualizing the Data in R

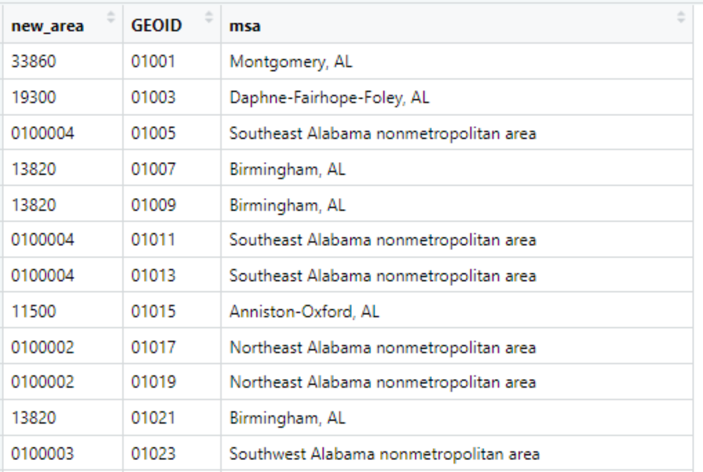
Visualization

3. I want a map, to highlight regional proximity in a way that separate bar charts do not.

OEWS Area Definitions



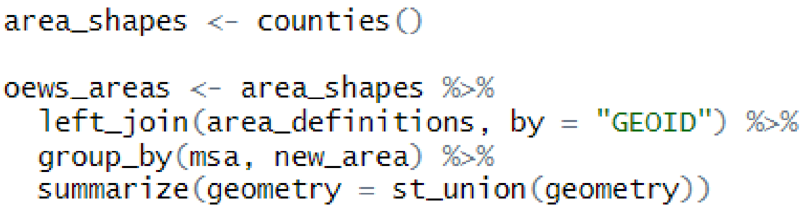
```
area_definitions <- read_excel("area_definitions_m2024.xlsx") %>%
  clean_names() %>%
  mutate(GEOID = paste0(fips,county_code)) %>%
  select(new_area, GEOID, msa)
```



new_area	GEOID	msa
33860	01001	Montgomery, AL
19300	01003	Daphne-Fairhope-Foley, AL
0100004	01005	Southeast Alabama nonmetropolitan area
13820	01007	Birmingham, AL
13820	01009	Birmingham, AL
0100004	01011	Southeast Alabama nonmetropolitan area
0100004	01013	Southeast Alabama nonmetropolitan area
11500	01015	Anniston-Oxford, AL
0100002	01017	Northeast Alabama nonmetropolitan area
0100002	01019	Northeast Alabama nonmetropolitan area
13820	01021	Birmingham, AL
0100003	01023	Southwest Alabama nonmetropolitan area

County Shapefiles + Merge

1. counties() gets all US counties via tigris
2. Join to OEWS definitions with GEOID
3. Group data by area code (“new_area”) and MSA name and combine shapes with st_union



```
area_shapes <- counties()

oews_areas <- area_shapes %>%
  left_join(area_definitions, by = "GEOID") %>%
  group_by(msa, new_area) %>%
  summarize(geometry = st_union(geometry))
```

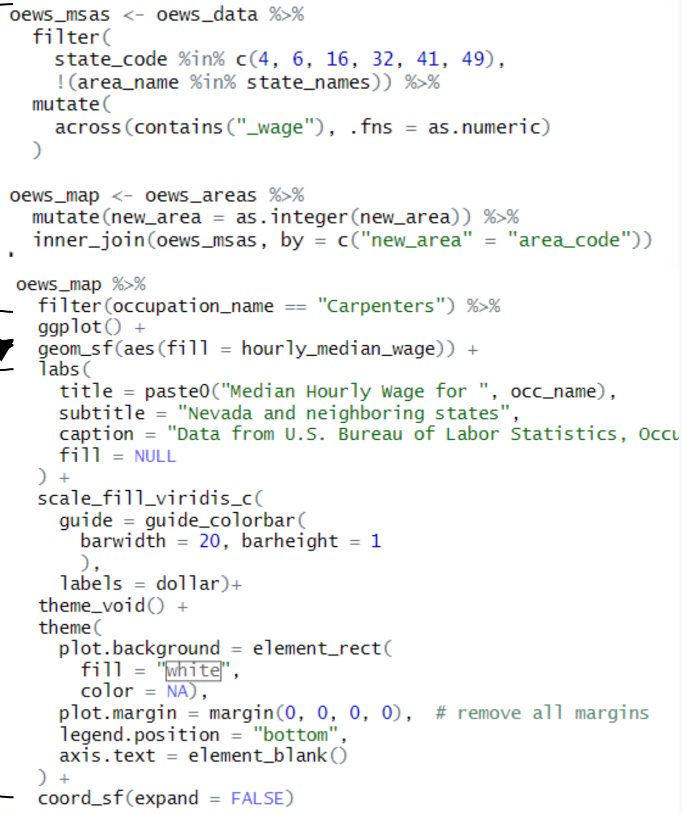
Visualizing the Data in R

Visualization

3. I want a map, to highlight regional proximity in a way that separate bar charts do not.

Elements of a ggplot:

1. Data set – what are you plotting?
2. Mapping (aesthetics) – what data columns will define your dynamic variables – x, y, color, fil, size, etc? Uses aes()
3. Geometries – what shape will that mapped data take? Uses geom_()
4. Scales, Labels, and Beautification

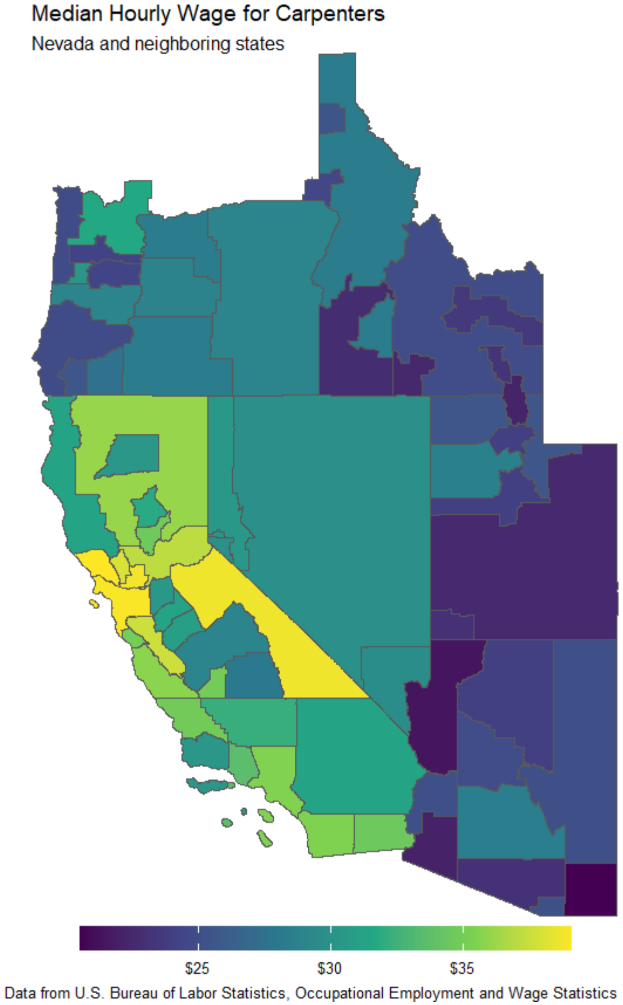


```
oews_msas <- oews_data %>%
  filter(
    state_code %in% c(4, 6, 16, 32, 41, 49),
    !(area_name %in% state_names)) %>%
  mutate(
    across(contains("_wage"), .fns = as.numeric)
  )

oews_map <- oews_areas %>%
  mutate(new_area = as.integer(new_area)) %>%
  inner_join(oews_msas, by = c("new_area" = "area_code"))
.
 oews_map %>%
   filter(occupation_name == "Carpenters") %>%
   ggplot() +
   geom_sf(aes(fill = hourly_median_wage)) +
   labs(
     title = paste0("Median Hourly Wage for ", occ_name),
     subtitle = "Nevada and neighboring states",
     caption = "Data from U.S. Bureau of Labor Statistics, Occu
     fill = NULL
   ) +
   scale_fill_viridis_c(
     guide = guide_colorbar(
       barwidth = 20, barheight = 1
       ),
     labels = dollar)+
   theme_void() +
   theme(
     plot.background = element_rect(
       fill = "white",
       color = NA),
     plot.margin = margin(0, 0, 0, 0),  # remove all margins
     legend.position = "bottom",
     axis.text = element_blank()
   ) +
   coord_sf(expand = FALSE)
```

Visualizing the Data in R

Visualization

We have all three visualizations! Investing in cleaning the data makes different variations on visualization much easier!


Median Hourly Wage for Carpenters

Nevada and neighboring states

Data from U.S. Bureau of Labor Statistics, Occupational Employment and Wage Statistics

But that was a lot of code that specifically references “Carpenters”. It’s hard-coded as a filter and typed in as the title. How do we scale this up? Right now, we’re not doing much more than Excel can easily do.

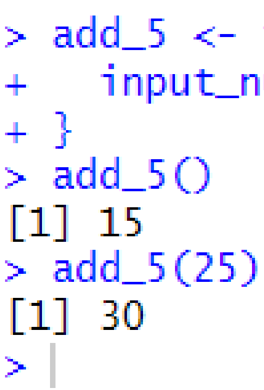
To scale up, we want to be able to iterate our analysis and to integrate those outputs into a finished product.

Iterating with Functions

If you are copying and pasting code, you should write a function!

1. As you tweak your code, using a function means you only make the change in one place, not every copy of it.
2. It’s built to apply the same logic to different inputs.
3. It’s easier than you think!

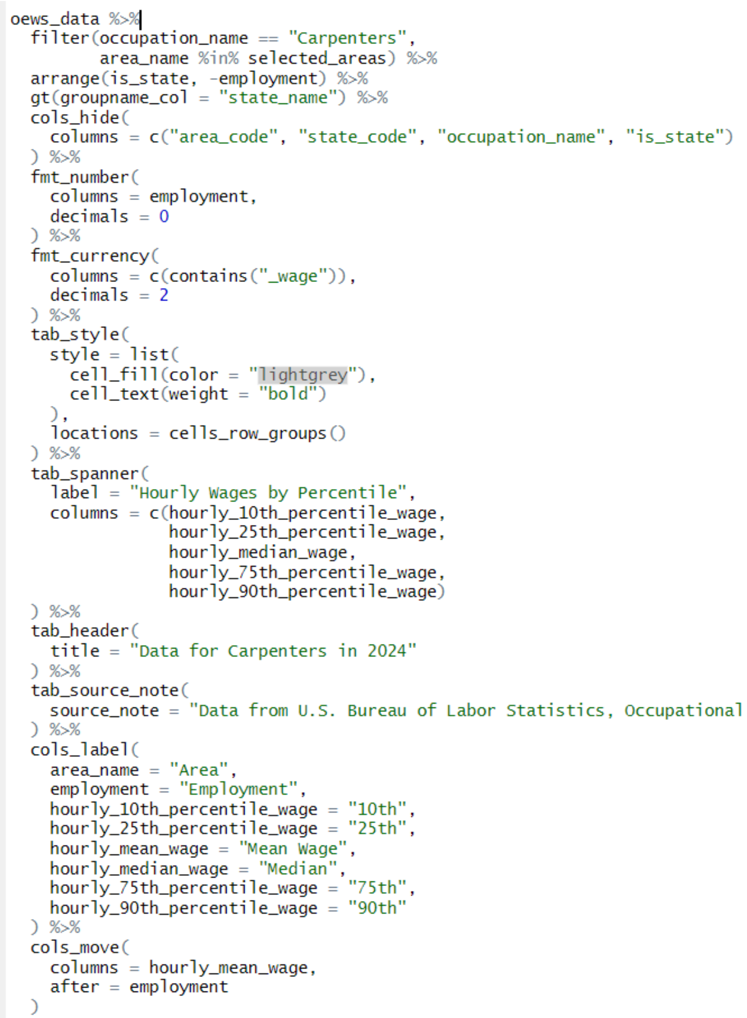
function_name <- **function(**variable_name = default_value, …**){**
****what it should do****
}



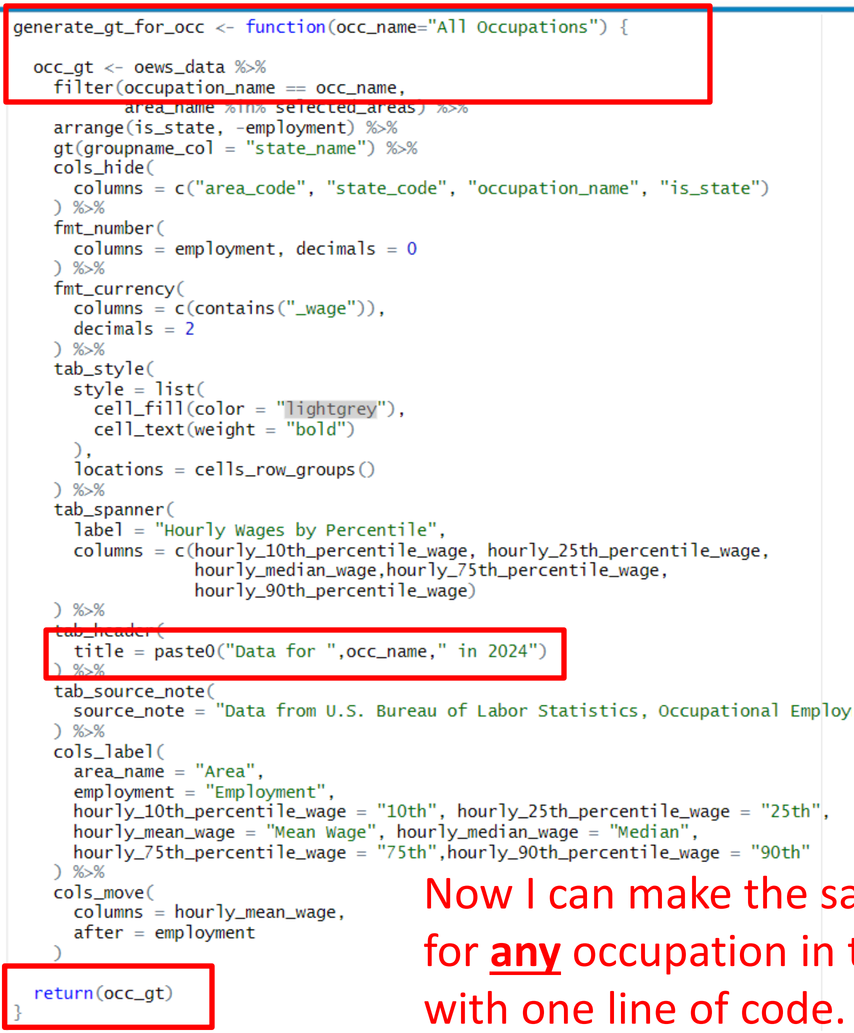
```
> add_5 <- function(input_number = 10){
+   input_number + 5
+ }
> add_5()
[1] 15
> add_5(25)
[1] 30
> 
```

Creating a function called add_5 with a default value of 10. When called, it takes the input_number and adds 5. If no input_number is provided, it uses a default of 10.

gt table: Original … and Function-al



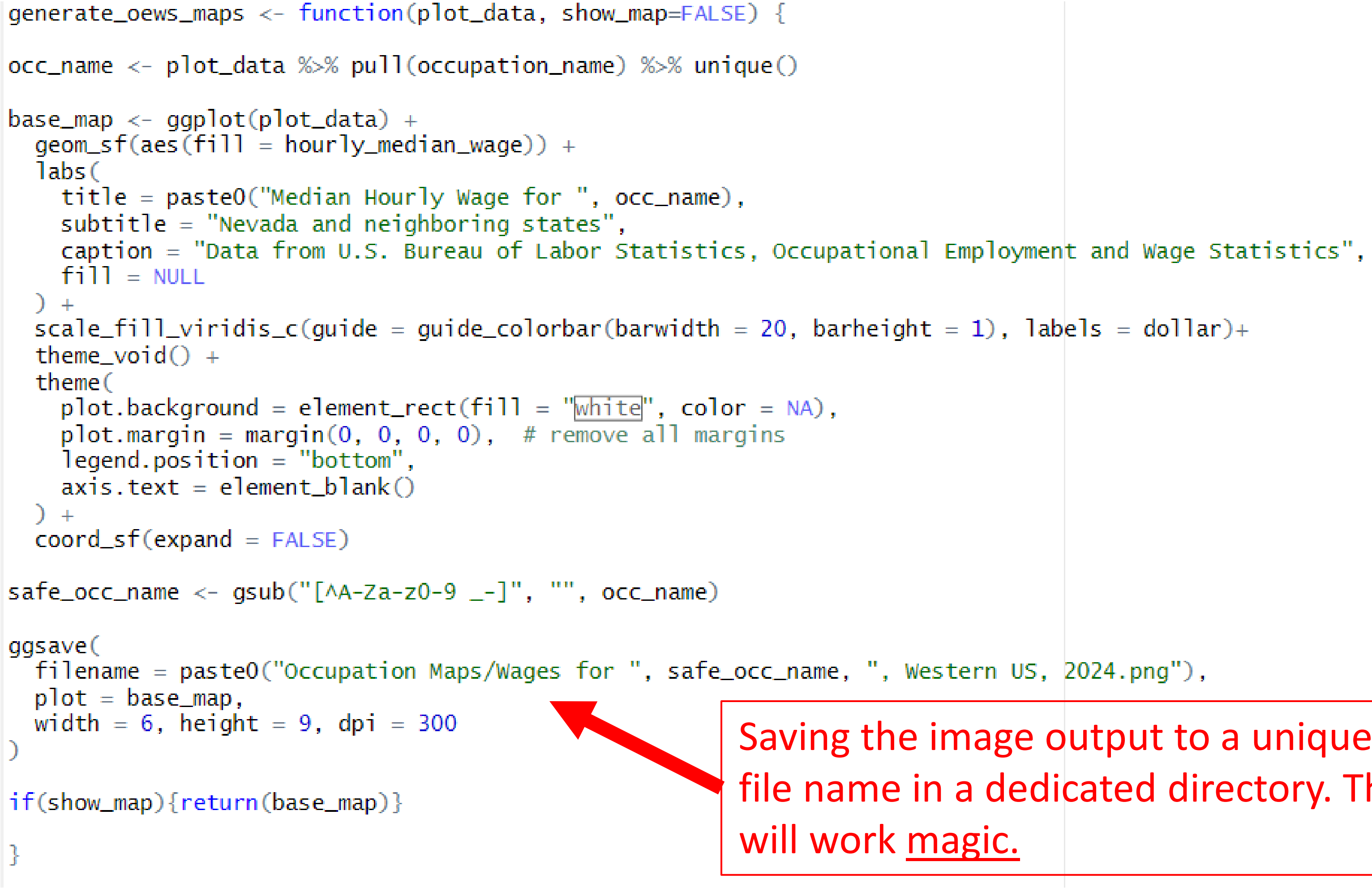
```
oews_data %>%
  filter(occupation_name == "Carpenters",
         area_name %in% selected_areas) %>%
  arrange(is_state, -employment) %>%
  gt(groupname_col = "state_name") %>%
  cols_hide(
    columns = c("area_code", "state_code", "occupation_name", "is_state")
  ) %>%
  fmt_number(
    columns = employment,
    decimals = 0
  ) %>%
  fmt_currency(
    columns = c(contains("_wage")),
    decimals = 2
  ) %>%
  tab_style(
    style = list(
      cell_fill(color = "lightgrey"),
      cell_text(weight = "bold")
    ),
    locations = cells_row_groups()
  ) %>%
  tab_spanner(
    label = "Hourly Wages by Percentile",
    columns = c(hourly_10th_percentile_wage,
                hourly_25th_percentile_wage,
                hourly_median_wage,
                hourly_75th_percentile_wage,
                hourly_90th_percentile_wage)
  ) %>%
  tab_header(
    title = "Data for Carpenters in 2024"
  ) %>%
  tab_source_note(
    source_note = "Data from U.S. Bureau of Labor Statistics, Occupational
  ) %>%
  cols_label(
    area_name = "Area",
    employment = "Employment",
    hourly_10th_percentile_wage = "10th",
    hourly_25th_percentile_wage = "25th",
    hourly_mean_wage = "Mean Wage",
    hourly_median_wage = "Median",
    hourly_75th_percentile_wage = "75th",
    hourly_90th_percentile_wage = "90th"
  ) %>%
  cols_move(
    columns = hourly_mean_wage,
    after = employment
  )
```



```
generate_gt_for_occ <- function(occ_name="All Occupations") {

  occ_gt <- oews_data %>%
    filter(occupation_name == occ_name,
           area_name %in% selected_areas) %>%
    arrange(is_state, -employment) %>%
    gt(groupname_col = "state_name") %>%
    cols_hide(
      columns = c("area_code", "state_code", "occupation_name", "is_state")
    ) %>%
    fmt_number(
      columns = employment, decimals = 0
    ) %>%
    fmt_currency(
      columns = c(contains("_wage")),
      decimals = 2
    ) %>%
    tab_style(
      style = list(
        cell_fill(color = "lightgrey"),
        cell_text(weight = "bold")
      ),
      locations = cells_row_groups()
    ) %>%
    tab_spanner(
      label = "Hourly Wages by Percentile",
      columns = c(hourly_10th_percentile_wage, hourly_25th_percentile_wage,
                  hourly_median_wage,hourly_75th_percentile_wage,
                  hourly_90th_percentile_wage)
    ) %>%
    tab_header(
      title = paste0("Data for ",occ_name," in 2024")
    ) %>%
    tab_source_note(
      source_note = "Data from U.S. Bureau of Labor Statistics, Occupational Employ
    ) %>%
    cols_label(
      area_name = "Area",
      employment = "Employment",
      hourly_10th_percentile_wage = "10th", hourly_25th_percentile_wage = "25th",
      hourly_mean_wage = "Mean Wage", hourly_median_wage = "Median",
      hourly_75th_percentile_wage = "75th",hourly_90th_percentile_wage = "90th"
    ) %>%
    cols_move(
      columns = hourly_mean_wage,
      after = employment
    )

  return(occ_gt)
}
```

Now I can make the same table for **any** occupation in the data with one line of code.

OEWS Mapping Function



```r
generate_oews_maps <- function(plot_data, show_map=FALSE) {

occ_name <- plot_data %>% pull(occupation_name) %>% unique()

base_map <- ggplot(plot_data) +
  geom_sf(aes(fill = hourly_median_wage)) +
  labs(
    title = paste0("Median Hourly Wage for ", occ_name),
    subtitle = "Nevada and neighboring states",
    caption = "Data from U.S. Bureau of Labor Statistics, Occupational Employment and Wage Statistics",
    fill = NULL
  ) +
  scale_fill_viridis_c(guide = guide_colorbar(barwidth = 20, barheight = 1), labels = dollar)+
  theme_void() +
  theme(
    plot.background = element_rect(fill = "white", color = NA),
    plot.margin = margin(0, 0, 0, 0),  # remove all margins
    legend.position = "bottom",
    axis.text = element_blank()
  ) +
  coord_sf(expand = FALSE)

safe_occ_name <- gsub("[^A-Za-z0-9 _-]", "", occ_name)

ggsave(
  filename = paste0("Occupation Maps/Wages for ", safe_occ_name, ", Western US, 2024.png"),
  plot = base_map,
  width = 6, height = 9, dpi = 300
)

if(show_map){return(base_map)}

}
```

Saving the image output to a unique file name in a dedicated directory. This will work <u>magic.</u>

Integrating with RMarkdown

If you want a report with multiple tables, maps, or plots – or if you simply want to include both text and outputs from R Code, it’s time to think about Rmarkdown or similar **rendering** options (e.g. Quarto).

R Script (*.R)

Processes the code. Good for processing content and saving *an* output to *a* directory.

COMBINED?

Create the report layout in RMD, then call it in a loop in a script to iterate across multiple inputs!

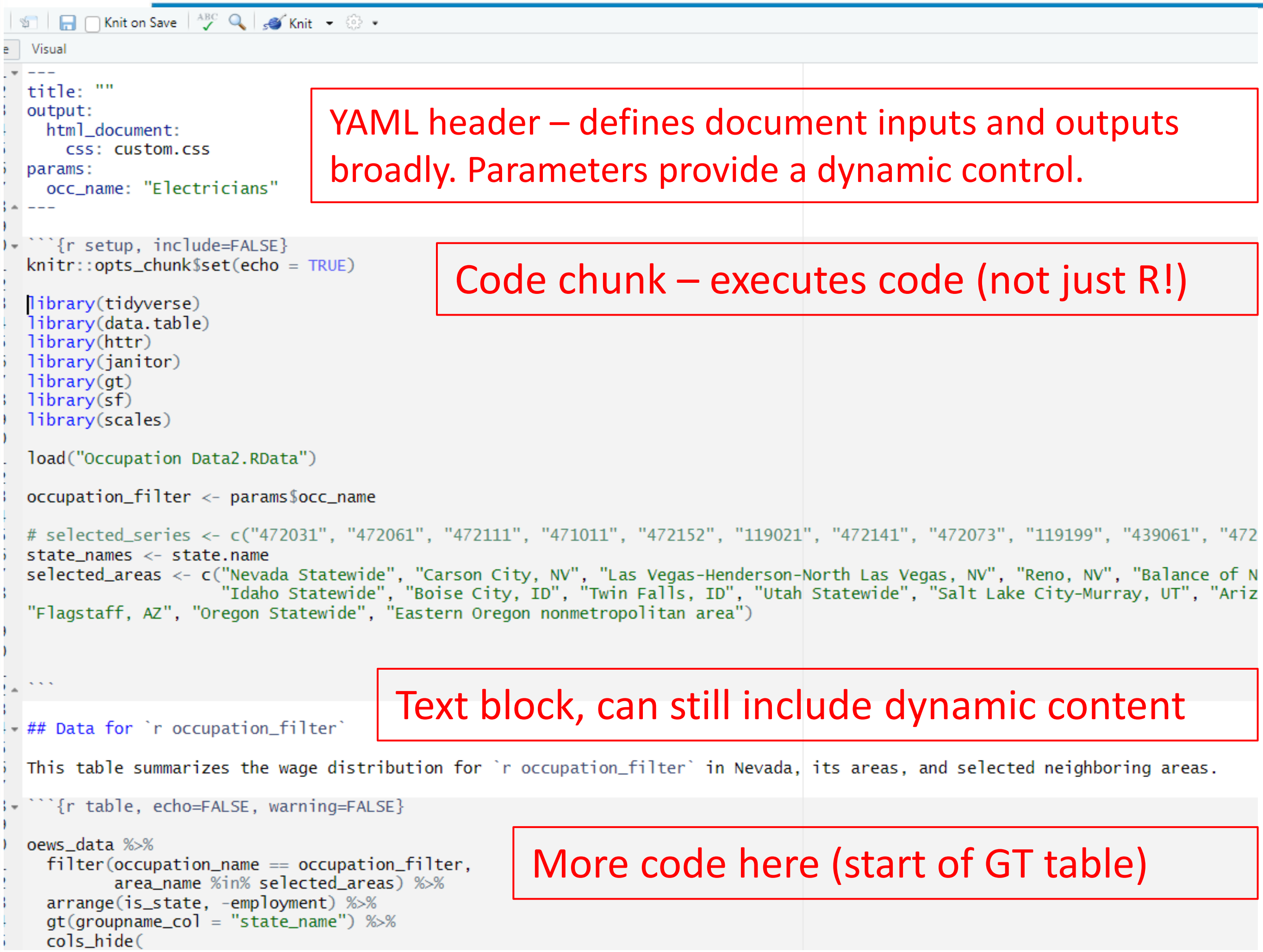
Rmarkdown (*.RMD)

Integrates multiple chunks of code, output, and text into a finished document, then combines it all into a single product. Text, tables, charts, and more.

Can do MS Office documents, HTML documents, PDF outputs, and more.

Runs code in isolated environment, and makes results more replicable by others.

Integrating with RMarkdown



~~~
---
title: ""
output:
  html_document:
    css: custom.css
params:
  occ_name: "Electricians"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

library(tidyverse)
library(data.table)
library(httr)
library(janitor)
library(gt)
library(sf)
library(scales)

load("Occupation Data2.RData")

occupation_filter <- params$occ_name

# selected_series <- c("472031", "472061", "472111", "471011", "472152", "119021", "472141", "472073", "119199", "439061", "472
state_names <- state.name
selected_areas <- c("Nevada Statewide", "Carson City, NV", "Las Vegas-Henderson-North Las Vegas, NV", "Reno, NV", "Balance of N
                    "Idaho Statewide", "Boise City, ID", "Twin Falls, ID", "Utah Statewide", "Salt Lake City-Murray, UT", "Ariz
"Flagstaff, AZ", "Oregon Statewide", "Eastern Oregon nonmetropolitan area")


~~~

Data for `r occupation_filter`

This table summarizes the wage distribution for `r occupation_filter` in Nevada, its areas, and selected neighboring areas.

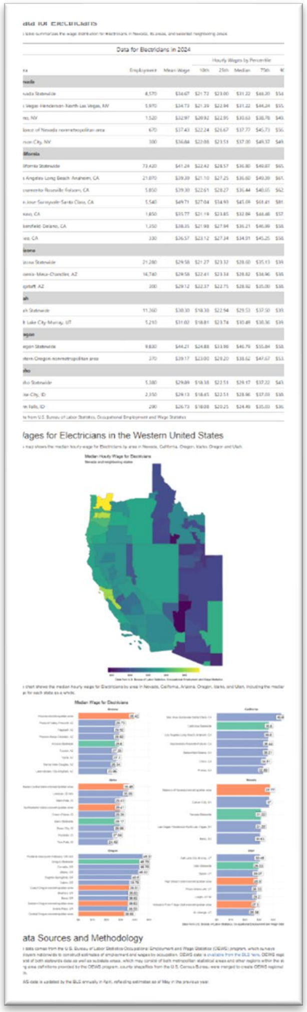
```{r table, echo=FALSE, warning=FALSE}

oews_data %>%
  filter(occupation_name == occupation_filter,
         area_name %in% selected_areas) %>%
  arrange(is_state, -employment) %>%
  gt(groupname_col = "state_name") %>%
  cols_hide(
```

YAML header – defines document inputs and outputs broadly. Parameters provide a dynamic control.

Code chunk – executes code (not just R!)

Text block, can still include dynamic content

More code here (start of GT table)


Data for Electricians in 2024
/ages for Electricians in the Western United States
ata Sources and Methodology

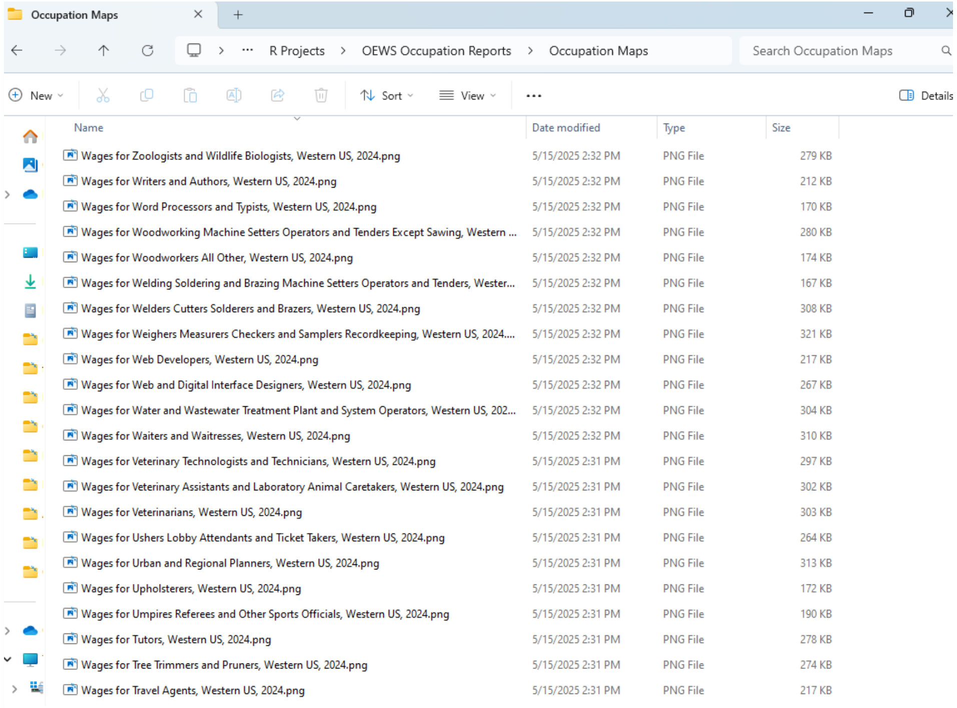
NevadaWorkforce.com

DETR
Nevada Department of Employment, Training and Rehabilitation

RESEARCH & ANALYSIS
Nevada's premier source of workforce and economic information and analysis

Don’t Be a Bad Computer!

You now have all the pieces to let the computer do the hard work. Don’t bother copying and pasting in every occupation – give your computer a list of occupations, and let it do all the work of pasting that list into your functions one at a time.



Name	Date modified	Type	Size
Wages for Zoologists and Wildlife Biologists, Western US, 2024.png	5/15/2025 2:32 PM	PNG File	279 KB
Wages for Writers and Authors, Western US, 2024.png	5/15/2025 2:32 PM	PNG File	212 KB
Wages for Word Processors and Typists, Western US, 2024.png	5/15/2025 2:32 PM	PNG File	170 KB
Wages for Woodworking Machine Setters Operators and Tenders Except Sawing, Western ...	5/15/2025 2:32 PM	PNG File	280 KB
Wages for Woodworkers All Other, Western US, 2024.png	5/15/2025 2:32 PM	PNG File	174 KB
Wages for Welding Soldering and Brazing Machine Setters Operators and Tenders, Wester...	5/15/2025 2:32 PM	PNG File	167 KB
Wages for Welders Cutters Solderers and Brazers, Western US, 2024.png	5/15/2025 2:32 PM	PNG File	308 KB
Wages for Weighers Measurers Checkers and Samplers Recordkeeping, Western US, 2024....	5/15/2025 2:32 PM	PNG File	321 KB
Wages for Web Developers, Western US, 2024.png	5/15/2025 2:32 PM	PNG File	217 KB
Wages for Web and Digital Interface Designers, Western US, 2024.png	5/15/2025 2:32 PM	PNG File	267 KB
Wages for Water and Wastewater Treatment Plant and System Operators, Western US, 202...	5/15/2025 2:32 PM	PNG File	304 KB
Wages for Waiters and Waitresses, Western US, 2024.png	5/15/2025 2:32 PM	PNG File	310 KB
Wages for Veterinary Technologists and Technicians, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	297 KB
Wages for Veterinary Assistants and Laboratory Animal Caretakers, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	302 KB
Wages for Veterinarians, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	303 KB
Wages for Ushers Lobby Attendants and Ticket Takers, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	264 KB
Wages for Urban and Regional Planners, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	313 KB
Wages for Upholsterers, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	172 KB
Wages for Umpires Referees and Other Sports Officials, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	190 KB
Wages for Tutors, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	278 KB
Wages for Tree Trimmers and Pruners, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	274 KB
Wages for Travel Agents, Western US, 2024.png	5/15/2025 2:31 PM	PNG File	217 KB

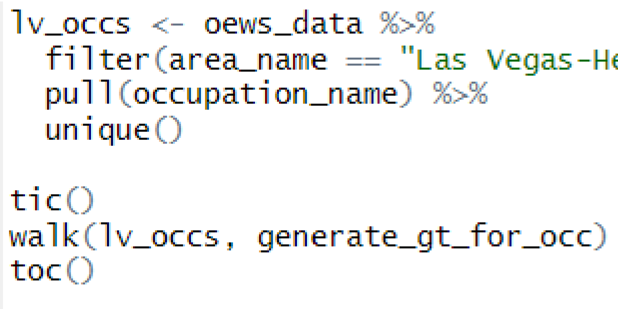
There are 628 occupations in the May 2024 OEWS data for Las Vegas, Nevada.

With a function in place, all I need to do is identify my list of occupation names in Las Vegas and then tell R to apply that function to every item in the list.

Scaling up with Iteration

If you build the process to generate your content correctly, the process of iterating is very simple – here, it takes just a couple lines of code.

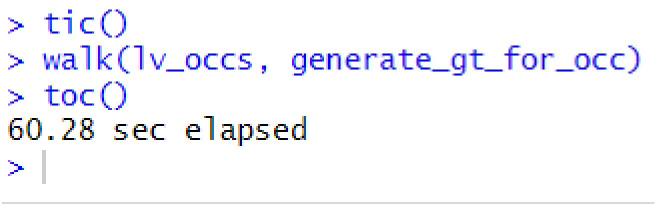
This is all it takes:



```
lv_occs <- oews_data %>%
  filter(area_name == "Las Vegas-Henderson-North Las Vegas, NV") %>%
  pull(occupation_name) %>%
  unique()

tic()
walk(lv_occs, generate_gt_for_occ)
toc()
```

Define what you want to serve as the inputs (here, unique occupation names in the Las Vegas area).

Use a function from the purrr package to apply the same function to a list of inputs. [Here using walk()]



```
> tic()
> walk(lv_occs, generate_gt_for_occ)
> toc()
60.28 sec elapsed
> |
```

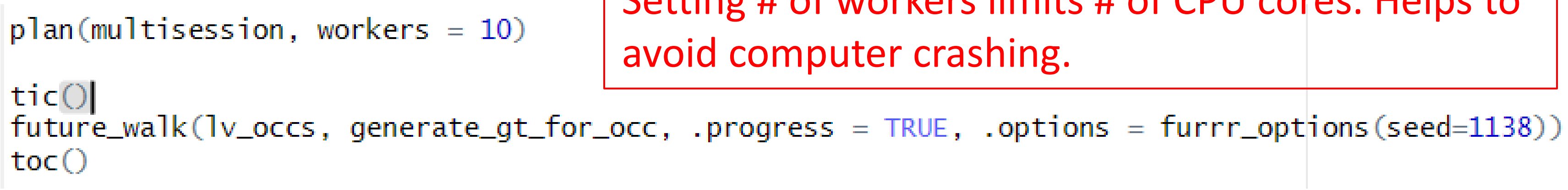
It took the computer only 60 seconds to generate a GT table for every single occupation – the tic() and toc() give me timing benchmarks.

Don’t get bogged down!

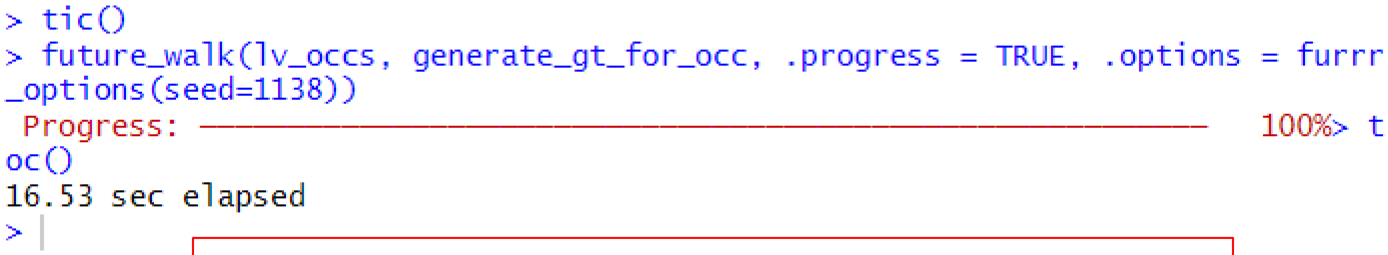
BUT… by default, functions like walk() and map() don’t use all your computer’s resources well. They don’t use multiple CPU cores, and so everything is happening in a line. It’s time to speed up and run this thing in parallel. For this, we’ll use the furrr package.

https://furrr.futureverse.org/

Setting # of workers limits # of CPU cores. Helps to avoid computer crashing.



```
plan(multisession, workers = 10)

tic()
future_walk(lv_occs, generate_gt_for_occ, .progress = TRUE, .options = furrr_options(seed=1138))
toc()
```



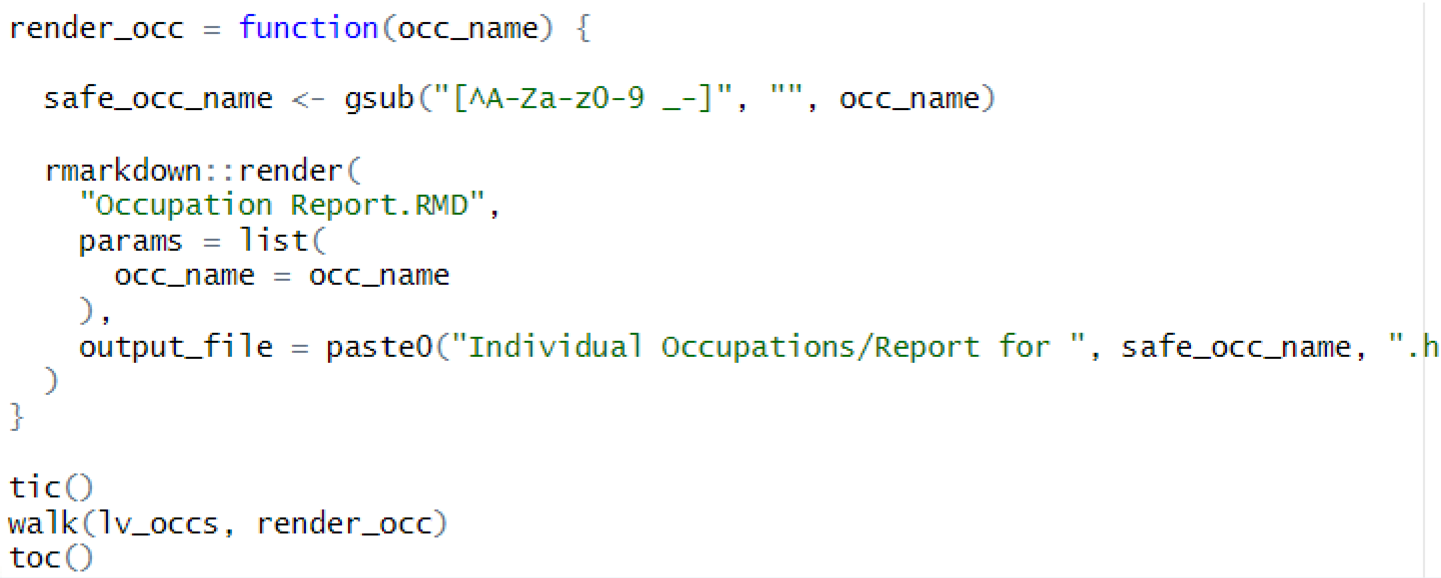
```
> tic()
> future_walk(lv_occs, generate_gt_for_occ, .progress = TRUE, .options = furrr
_options(seed=1138))
 Progress: ──────────────────────────────────────────────────   100%> t
oc()
16.53 sec elapsed
> 
```

Roughly 4x performance improvement!

Combining Script and Markdown

Finally, let's go back to our Markdown (RMD) document. We're now going to use that script inside another script. We're going to pass our list of occupations into the parameter in the YAML header to generate that report for every occupation in our list, and save those outputs each with their own unique file name.

This takes longer, but critically – it's time your computer is working hard, but you can be doing something else.



```
render_occ = function(occ_name) {

  safe_occ_name <- gsub("[^A-Za-z0-9 _-]", "", occ_name)

  rmarkdown::render(
    "Occupation Report.RMD",
    params = list(
      occ_name = occ_name
    ),
    output_file = paste0("Individual Occupations/Report for ", safe_occ_name, ".html")
  )
}

tic()
walk(lv_occs, render_occ)
toc()
```

But Why Would I Do This?

1. Eliminating the routine tasks frees you up to focus on analysis.

Don't waste time being a bad computer.

2. Manual process restrict our capacity and put blinders on what we look at. Iterating through all the data broadens our perspective.

What was important yesterday may not be important tomorrow.

3. In times of diminishing funding, achieving efficiency is necessary to continue meeting the data needs we face.

Get busy living or get busy dying.

I Need More!

Learning R is much easier using GenAI:
https://chatgpt.com/share/681ba433-af94-800c-a557-b5234f3e2929

Following R users on social media is a great prompt for ideas:
https://www.linkedin.com/in/walkerke/

There are a lot of free resources available to watch and replicate code:
https://www.youtube.com/watch?v=9a8_p_q4Z34&t=795s
https://www.youtube.com/watch?v=8NKj8yF2gfo
https://www.youtube.com/watch?v=4WZfw0K7Vx8

I help lead a monthly R user group focused on state workforce agencies. Sign up here >>>

Get code I used today from Github:
https://github.com/schmidtDETR/OEWS-Mapping-and-Visualization