



Analyzing United States Census Bureau's December 2017 Basic Monthly CPS

By:
Paulina John



According to the [US Bureau of Labour Statistics website](#), the monthly CPS (Current Population Survey) data is a monthly survey of households conducted by the Bureau of Census for the Bureau of Labor Statistics and It provides a comprehensive body of data on employment and unemployment, the labour force, earnings and hours of work, other demographic, amongst of other details that relate to people in the US

I answered some questions on the December 2017 data and the following slides are snapshots of the results

Question 1: What is the count of responders per family income range (show all)?

To do this, we group by family_income_range and take the count of responders in each income range. First, let's exclude invalid entries

```
✓ 9s [12] cps_incomerange_valid = cps_2017_df.filter(col("family_income_range") != "INVALID ENTRY")  
  
      family_income_range_count = cps_incomerange_valid.groupBy("family_income_range").count()  
  
      family_income_range_count.show(truncate=False) # To show all, truncate=False
```

```
+-----+-----+  
|family_income_range|count|  
+-----+-----+  
|$35,000 TO $39,999|6620|  
|$5,000 TO $7,499  |1625|  
|$30,000 TO $34,999|6743|  
|$7,500 TO $9,999  |2277|  
|$25,000 TO $29,999|5803|  
|$20,000 TO $24,999|6312|  
|$10,000 TO $12,499|3161|  
|$50,000 TO $59,999|9971|  
|$40,000 TO $49,999|9788|  
|LESS THAN $5,000  |3136|  
|$12,500 TO $14,999|2614|  
|$75,000 TO $99,999|16557|  
|$60,000 TO $74,999|13442|  
|$100,000 TO $149,999|17794|  
|$150,000 OR MORE   |15704|  
|$15,000 TO $19,999|4518|  
+-----+-----+
```

Question 2: What is the count of responders per geographical division/location and race

Here, we group by both geographical division/location and race and then we count. We also first filter out invalid entries.

✓
7s



```
from pyspark.sql import functions as F
```

```
division_race_valid = cps_2017_df.filter((col("geographical_division") != "INVALID ENTRY") & (col("race") != "INVALID ENTRY"))
```

```
division_race_count = division_race_valid.groupBy("geographical_division", "race").count()
```

```
division_race_count_top_10 = division_race_count.orderBy(F.desc("count")).limit(10)
```

```
division_race_count_top_10.show()
```

```
+-----+-----+-----+
|geographical_division|      race|count|
+-----+-----+-----+
|      SOUTH ATLANTIC|White Only|16999|
|      MOUNTAIN|White Only|14343|
|      PACIFIC|White Only|13214|
|EAST NORTH CENTRAL|White Only|11325|
|WEST SOUTH CENTRAL|White Only|11248|
|WEST NORTH CENTRAL|White Only| 9884|
|  MIDDLE ATLANTIC|White Only| 8487|
|    NEW ENGLAND|White Only| 8410|
|EAST SOUTH CENTRAL|White Only| 6580|
|      SOUTH ATLANTIC|Black Only| 4899|
+-----+-----+-----+
```

Question 3: How many responders do not have telephone in their house, but can access a telephone elsewhere and telephone interview is accepted?

✓
7s

```
[▶] df_filtered = cps_2017_df.filter(      # Not forgetting to filter out invalid entries
    (col("telephone_in_household") != "INVALID ENTRY") &
    (col("telephone_accessible_elsewhere") != "INVALID ENTRY") &
    (col("telephone_interview_acceptable") != "INVALID ENTRY")
)

telephone_responders_filtered = df_filtered.filter(
    (col("telephone_in_household") == "NO") &
    (col("telephone_accessible_elsewhere") == "YES") &
    (col("telephone_interview_acceptable") == "YES")
)

# Computing the count
responders_count = telephone_responders_filtered.count()

# Printing result
print(responders_count, "responders do not have a telephone in their house but can access elsewhere and telephone interview is accepted:")
```

👉 633 responders do not have a telephone in their house but can access elsewhere and telephone interview is accepted:

Question 4: How many responders can access a telephone, but telephone interview is not accepted?__



```
filtered_df = cps_2017_df.filter(  
    (col("telephone_in_household") != "INVALID ENTRY") &  
    (col("telephone_accessible_elsewhere") != "INVALID ENTRY") &  
    (col("telephone_interview_acceptable") != "INVALID ENTRY")  
)  
  
filtered_responders = filtered_df.filter(  
    (col("telephone_in_household") == "YES") |  
    (col("telephone_accessible_elsewhere") == "YES") &  
    (col("telephone_interview_acceptable") == "NO")  
)  
  
num_of_responders = filtered_responders.count()  
  
print(num_of_responders, "responders can access a telephone but telephone interview is not accepted:")
```

4231 responders can access a telephone but telephone interview is not accepted:

Resources Used



- Google search engine
- https://www.bd-econ.com/nbs/cps_read_basic.html
- <https://www.census.gov/data/datasets/2017/demo/cps/cps-basic-2017.html>
- ChatGPT



Thank You