```python
# -*- coding: UTF-8 -*-
"""PyPoll Homework Challenge Solution."""

# Add our dependencies.
import csv
import os

# Add a variable to load a file from a path.
file_to_load = os.path.join("Resources", "election_results.csv")
# Add a variable to save the file to a path.
file_to_save = os.path.join("analysis", "election_analysis.txt")

# Initialize a total vote counter.
total_votes = 0

# Candidate Options and candidate votes.
candidate_options = []
candidate_votes = {}

# 1: Create a county list and county votes dictionary.
county_options = []
county_votes = {}


# Track the winning candidate, vote count and percentage
winning_candidate = ""
winning_count = 0
winning_percentage = 0

# 2: Track the largest county and county voter turnout.
winning_county=""
winning_county_turnout=0


# Read the csv and convert it into a list of dictionaries
with open(file_to_load) as election_data:
    reader = csv.reader(election_data)

    # Read the header
    header = next(reader)

    # For each row in the CSV file.
    for row in reader:

        # Add to the total vote count
        total_votes = total_votes + 1

        # Get the candidate name from each row.
        candidate_name = row[2]

        # 3: Extract the county name from each row.
        county_name = row[1]

        # If the candidate does not match any existing candidate add it to
        # the candidate list
        if candidate_name not in candidate_options:

            # Add the candidate name to the candidate list.
            candidate_options.append(candidate_name)

            # And begin tracking that candidate's voter count.
            candidate_votes[candidate_name] = 0

        # Add a vote to that candidate's count
        candidate_votes[candidate_name] += 1

        # 4a: Write an if statement that checks that the
        # county does not match any existing county in the county list.
        if county_name not in county_options:
            # 4b: Add the existing county to the list of counties.
            county_options.append(county_name)
            # 4c: Begin tracking the county's vote count.
            county_votes[county_name]=0
        # 5: Add a vote to that county's vote count.
        county_votes[county_name]+=1

# Save the results to our text file.
with open(file_to_save, "w") as txt_file:

    # Print the final vote count (to terminal)
    election_results = (
        f"\nElection Results\n"
        f"-------------------------\n"
        f"Total Votes: {total_votes:,}\n"
        f"-------------------------\n\n"
        f"County Votes:\n")
    print(election_results, end="")

    txt_file.write(election_results)

    # 6a: Write a for loop to get the county from the county dictionary. (county_votes is the name of the dictionary)
    for i in county_votes:
        # 6b: Retrieve the county vote count.
        number_of_county_votes=county_votes.get(i)
        # 6c: Calculate the percentage of votes for the county.
        county_vote_percentage=float(number_of_county_votes)/float(total_votes)*100
         # 6d: Print the county results to the terminal.
        county_results= (f"{i}: {county_vote_percentage:.1f}% ({number_of_county_votes:,})\n")
        print(county_results)
         # 6e: Save the county votes to a text file.
        txt_file.write(county_results)
         # 6f: Write an if statement to determine the winning county and get its vote count.
        if number_of_county_votes > winning_county_turnout:
            winning_county_turnout=number_of_county_votes
            winning_county=i
    # 7: Print the county with the largest turnout to the terminal.
    largest_turnout_county_details= (
        f"\n-------------------------\n"
        f"Largest County Turnout: {winning_county}\n"
        f"-------------------------\n")
    print(largest_turnout_county_details)

    # 8: Save the county with the largest turnout to a text file.
    txt_file.write(largest_turnout_county_details)

    # Save the final candidate vote count to the text file.
    for i in candidate_votes:

        # Retrieve vote count and percentage
        votes = candidate_votes.get(i)
        vote_percentage = float(votes) / float(total_votes) * 100
        candidate_results = (
            f"{i}: {vote_percentage:.1f}% ({votes:,})\n")

        # Print each candidate's voter count and percentage to the
        # terminal.
        print(candidate_results)
        #  Save the candidate results to our text file.
        txt_file.write(candidate_results)

        # Determine winning vote count, winning percentage, and candidate.
        if (votes > winning_count) and (vote_percentage > winning_percentage):
            winning_count = votes
            winning_candidate = i
            winning_percentage = vote_percentage

    # Print the winning candidate (to terminal)
    winning_candidate_summary = (
        f"-------------------------\n"
        f"Winner: {winning_candidate}\n"
        f"Winning Vote Count: {winning_count:,}\n"
        f"Winning Percentage: {winning_percentage:.1f}%\n"
        f"-------------------------\n")
    print(winning_candidate_summary)

    # Save the winning candidate's name to the text file
    txt_file.write(winning_candidate_summary)

```